



팀원소개

- 백엔드팀 -



허준



한국



김예

- 데이터팀 -



안준



김민



윤정

CONTENT

- 데이터팀 -

1부

1 주제선정 배경

- 서비스 소개
- 서비스 기대효과

2 데이터 수집 및 전처리

- 사용 데이터
- 데이터 시각화

3 모델선택과정

- 전통적 모델
- 머신/딥 러닝

4 모델 평가

- 모델 성능 측정 지표
- 모델 평가 결과

5 웹구현

- 웹 구현 과정

6 시연 동영상

- 시연

CONTENT

- 백엔드팀 -

2부

1 기술 스택

- 프론트엔드
- 백엔드
- 컴파일러

2 기능 소개

- 지도
- Security + JWT
- 커뮤니티

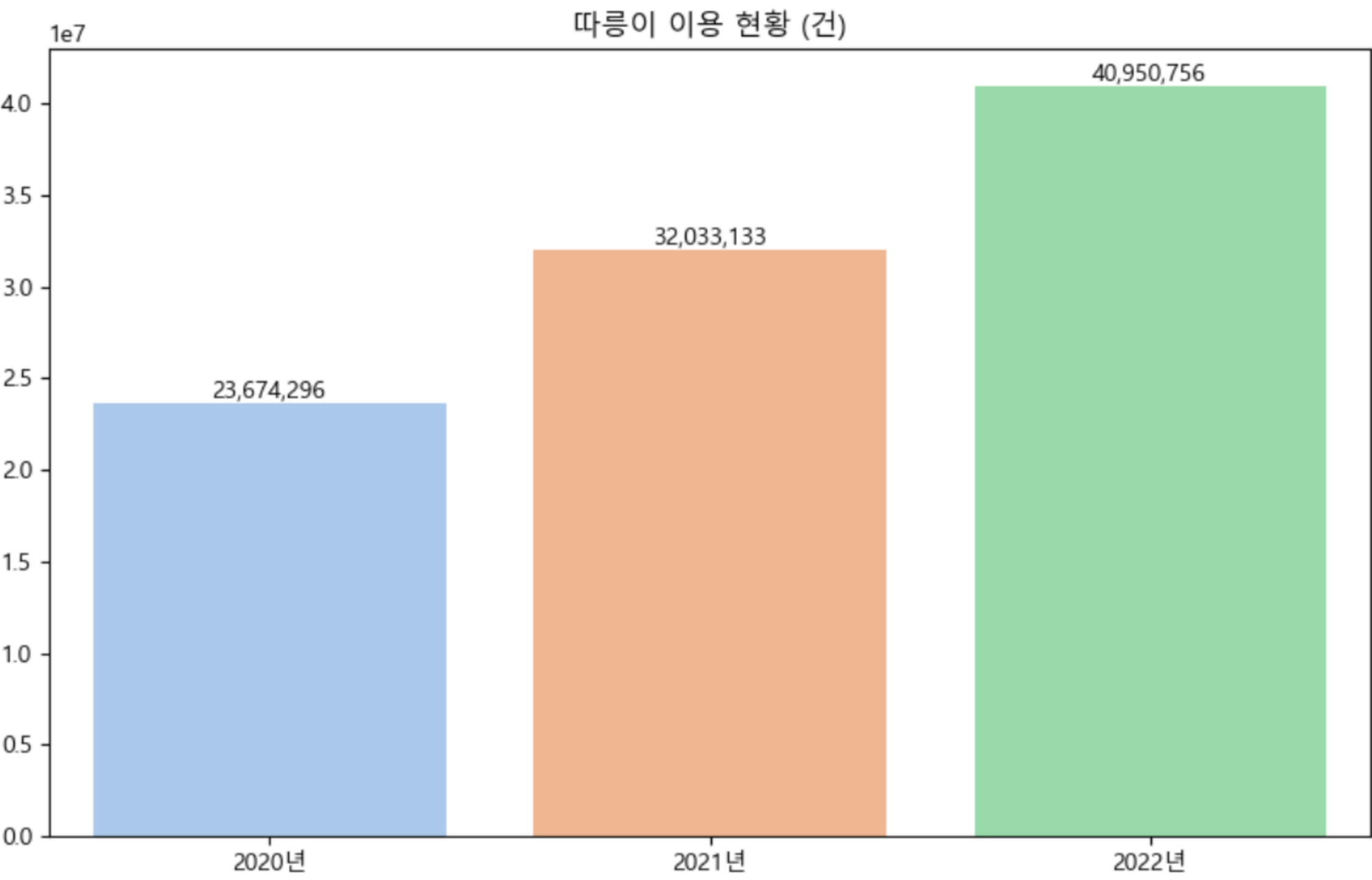
3 시연 동영상

- 시연

4 마무리

- 프로젝트 회고
- 보완할 사항

5 Q&A





따릉이

: 서울시에서 운영하는 공공자전거

자전거 : 41,000 대
대여소 : 2,756 개소



자전거
수요의 불균형



대여소에 도착했지만
잔여대수가 0대



[핵심]

“시민수요충족 및 안전확보”

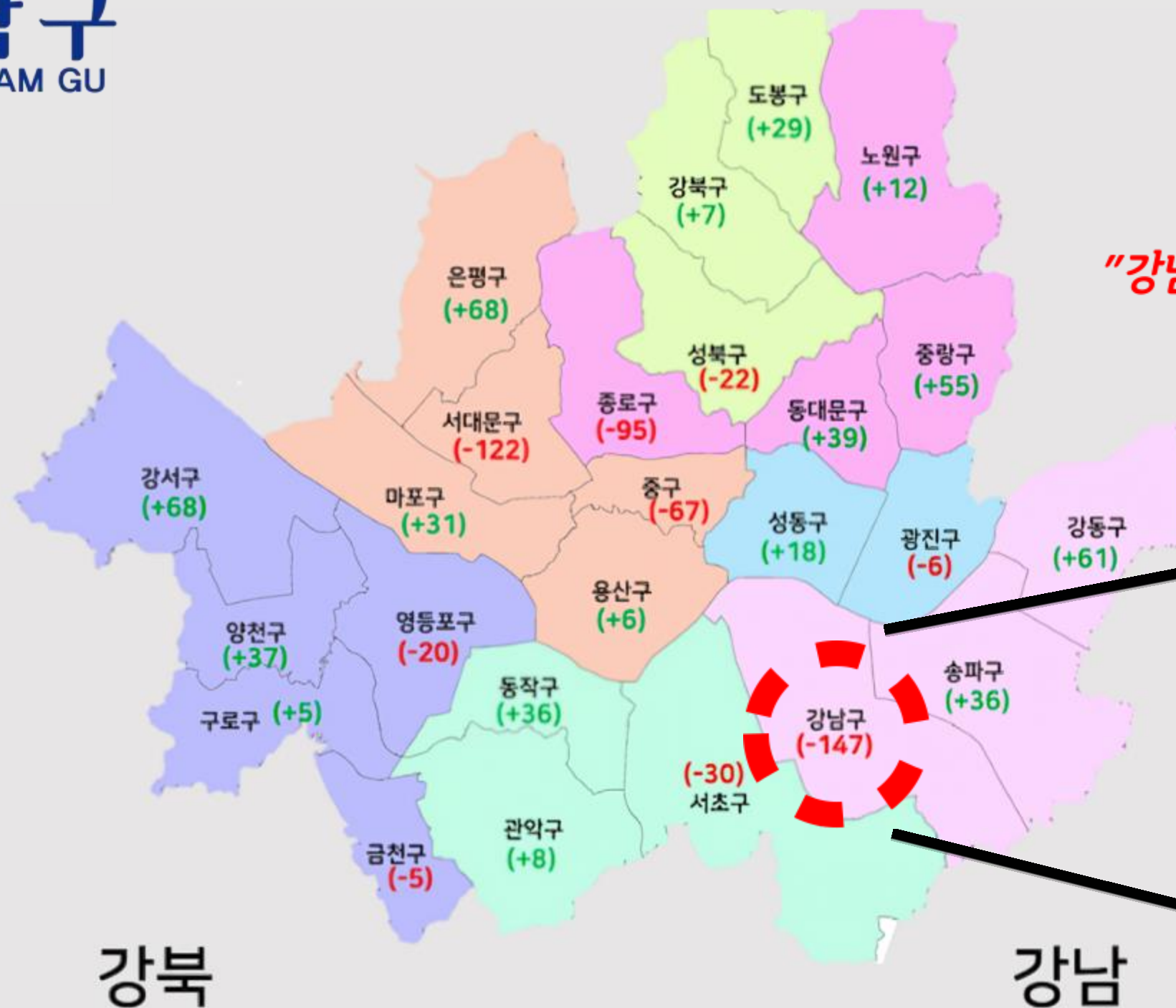
집중관리 대여소 확대

- 총 110개소 집중관리(11개소 선별 10개 대여소)
 - 빅데이터 거치 예측값 도출, 대여소 민원 분석
- 상수기 기간(3월~11월) 전담직원 편성

주제 선택

서울

델



"강남구"가 -147로 초과 대여
가 가장 많음

강남구
(-147)

- | | |
|-----------------------------|------------------------------|
| ● : 노원구, 중랑구, 동대문구, 종로구 | ● : 강동구, 송파구, 강남구 |
| ● : 도봉구, 강북구, 성북구 | ● : 서초구, 관악구, 동작구 |
| ● : 성동구, 광진구 | ● : 영등포구, 양천구, 금천구, 구로구, 강서구 |
| ● : 은평구, 서대문구, 마포구, 중구, 용산구 | |

서비스 소개

따릉이 예측 서비스, 나를 따랑



지도 화면



채팅방 화면

▶ 시간대별 예측 서비스를 통한 이용 편의성 증대

나를 따랑은 사용자가 원하는 대여소의 시간대별 따릉이의 잔여대수를 정확하게 예측함으로써 사용자가 보다 더 간편하게 따릉이를 이용할 수 있게 해주며, 지역별 따릉이 소모임 커뮤니티를 통한 지역 사회 활성화에 이바지합니다.

서비스 기대효과



나를 따랑은 사용자가 원하는 대여소의 시간대별 따릉이의 잔여대수를 정확하게 예측함으로써 사용자가 보다 더 간편하게 따릉이를 이용할 수 있게 해주며, 지역별 따릉이 소모임 커뮤니티를 통한 지역사회 활성화에 이바지합니다.

나랑따릉



	1주차 ▶ 주제선정			2주차 ▶ 데이터 전처리 및 환경 세팅			3주차 ▶ 데이터 전처리 및 웹 개발			4주차 ▶ 모델 학습 및 웹 개발		
업무 \ 일정	11.20	11.25	12.03	12.05	12.07	12.10	12.12	12.15	12.18	12.20	12.24	12.27
프로젝트 기획	주제 선정 및 환경 세팅											
데이터 수집 및 가공			데이터 전처리									
디자인 개발						웹 페이지 구현						
백엔드 개발						백엔드 기능 구현						
모델 학습						모델 설계 및 테스트						
테스트 및 완성										테스트 및 점검		

사용 데이터 - 대여소



[서울시 공공자전거 대여소 정보 - 공공자전거 대여소 정보\(22.12월 기준\).csv](#)

[서울시 따릉이대여소 마스터 정보](#)

[서울시 따릉이 대여소별 대여/반납 승객수 정보 - tpss_bcycl_od_statnhm_2020~2022.zip](#)

[서울시 공공자전거 실시간 대여정보\(API\)](#)

[서울시 공공자전거 대여 이력\(API\)](#)

기준_날짜	기준_시간	시작_대여소	종료_대여소	전체건수	전체_이용_ 분	전체_이용_ 거리
20220706	0	ST-1584	ST-487	1	26	5246
20220706	0	ST-410	ST-2268	1	89	3231
20220706	0	ST-395	ST-262	1	3	755
20220706	0	ST-350	ST-222	1	6	1320

서울시 따릉이 대여소별 대여/반납 승객수 정보 .csv

대여 대여소번호	대여 대여소명	대여대여소ID
2331	동영문화센터앞	ST-814
2372	대치역 사거리	ST-1181
2413	도곡역 1번 출구	ST-1879
2380	삼성동베이직하우스앞	ST-1245

공공자전거 대여소 정보(22.12월 기준).csv

사용 데이터 - 날씨

[서울시 대기오염 측정정보.csv](#)

[서울시 기온정보.csv](#)

[서울시 강수량정보.csv](#)



지점	지점명	일시	평균기온(°C)	일강수량(mm)
108	서울	2022-01-01	-4.3	0
108	서울	2022-01-02	-1.3	0
108	서울	2022-01-03	-1.9	0
108	서울	2022-01-04	-2.5	0

사용 데이터 - 유동인구

[행정동 단위 서울 생활인구\(내국인\).csv](#)

[서울시 역사마스터 정보.csv](#)

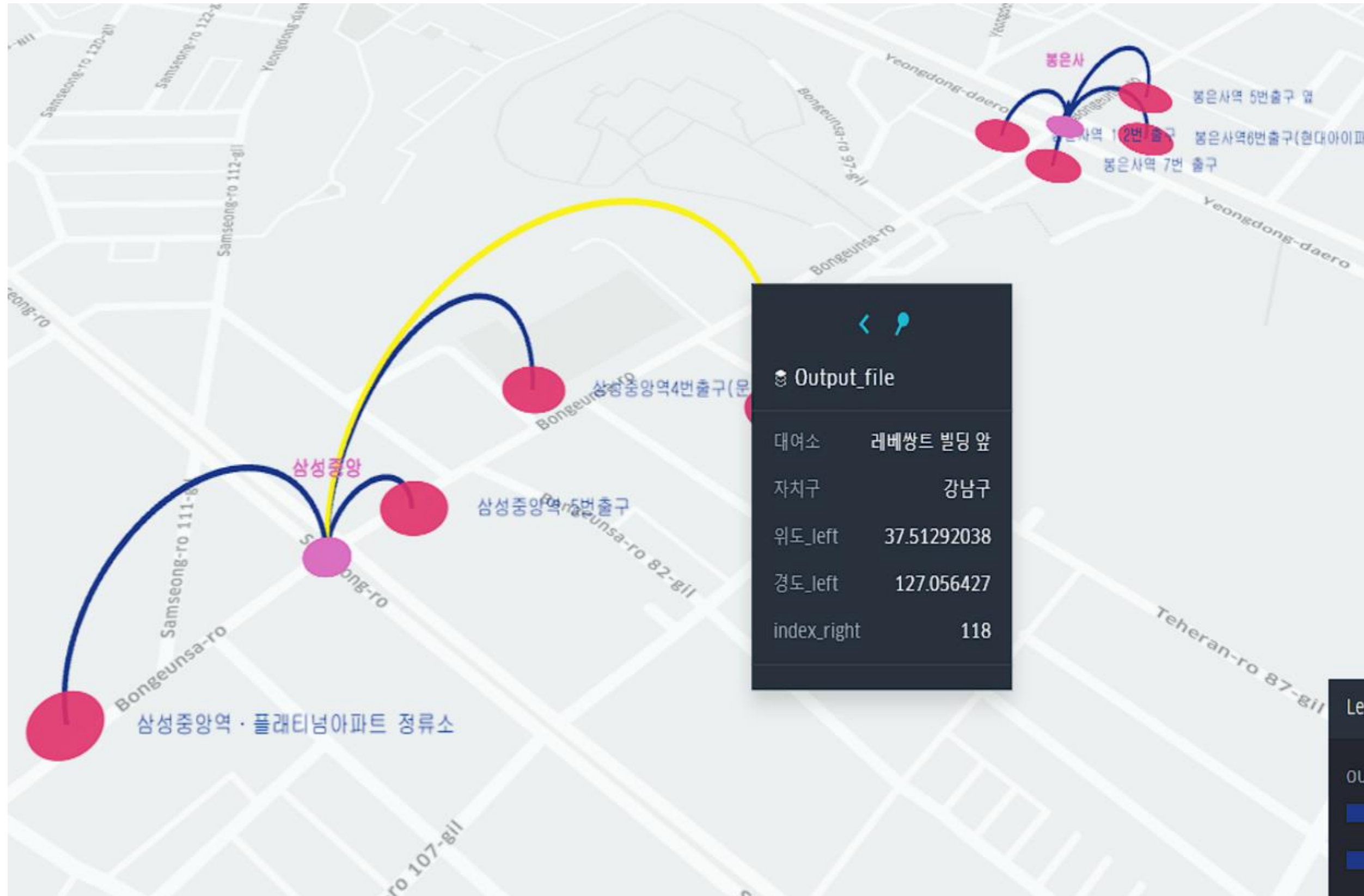
기준일ID	시간대구분	총생활인구수	행정동	대여소번호	보관소(대여소)명
20220101	0	26102.8887	대치4동	2325	대치삼성아파트 앞 교차로
20220101	0	47731.5554	세곡동	2343	세곡사거리 대왕파출소 앞
20220101	0	47731.5554	세곡동	2344	리엔파크 2단지 앞
20220101	0	47731.5554	세곡동	2383	보금자리정원

400m 데이터



역 400m 내 대여소 존재 여부

400m 데이터



역 400m 내 대여소 존재 여부

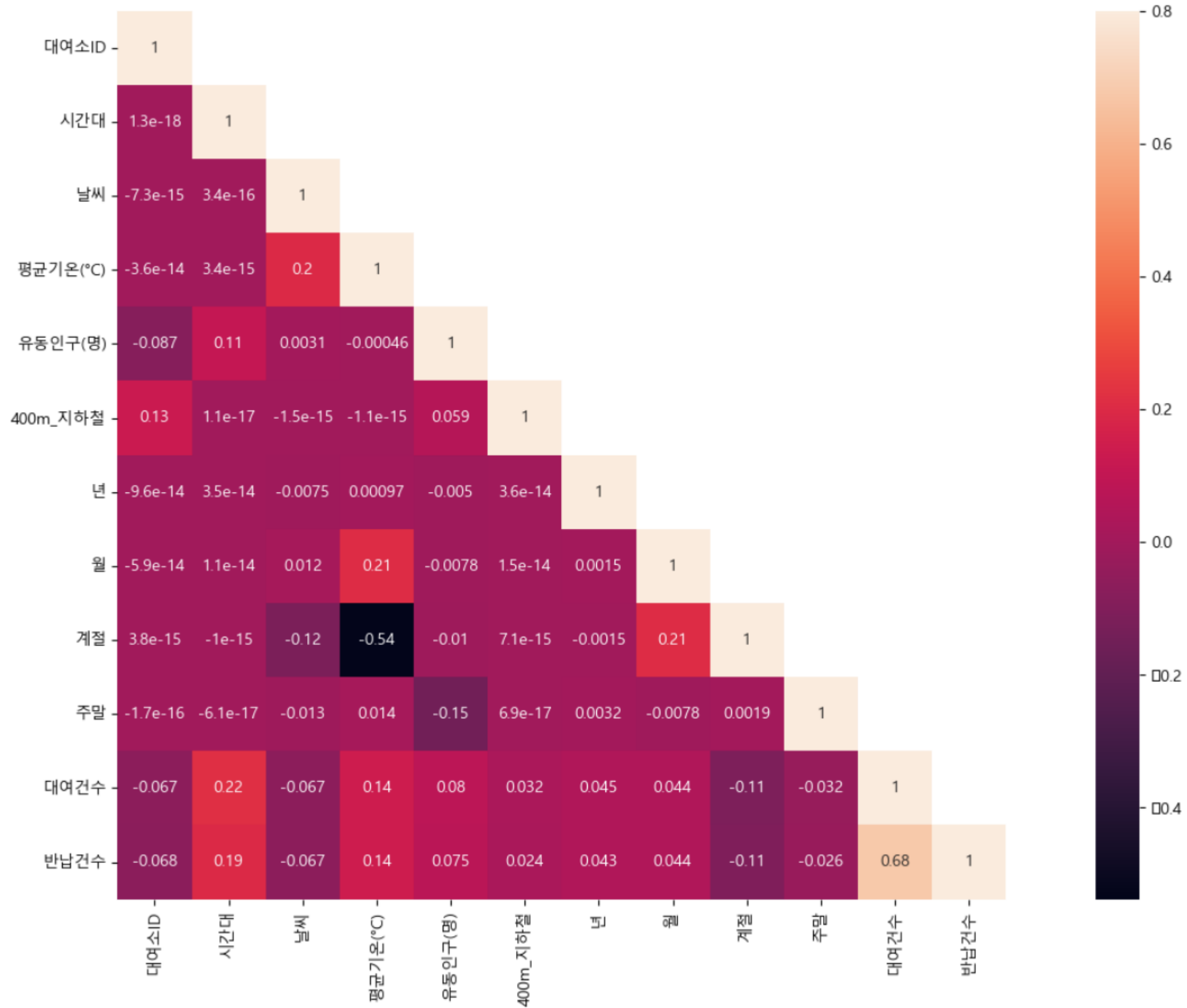
정제된 데이터



대여소ID	시간대	날씨	평균기온(°C)	유동인구(명)	PM10	대여건수	반납건수	요일	년	월	일	400m_지하철
ST-818	0	0	-4.3	29299	20	0	0	5	2022	1	1	1
ST-818	1	0	-4.3	29515	18	0	0	5	2022	1	1	1
ST-818	2	0	-4.3	29670	18	0	0	5	2022	1	1	1
ST-818	3	0	-4.3	29565	18	0	1	5	2022	1	1	1
ST-818	4	0	-4.3	29753	23	4	3	5	2022	1	1	1
ST-818	5	0	-4.3	30064	20	0	2	5	2022	1	1	1
ST-818	6	0	-4.3	29743	23	0	0	5	2022	1	1	1
ST-818	7	0	-4.3	29084	22	0	2	5	2022	1	1	1
ST-818	8	0	-4.3	28487	24	0	0	5	2022	1	1	1

상관계수

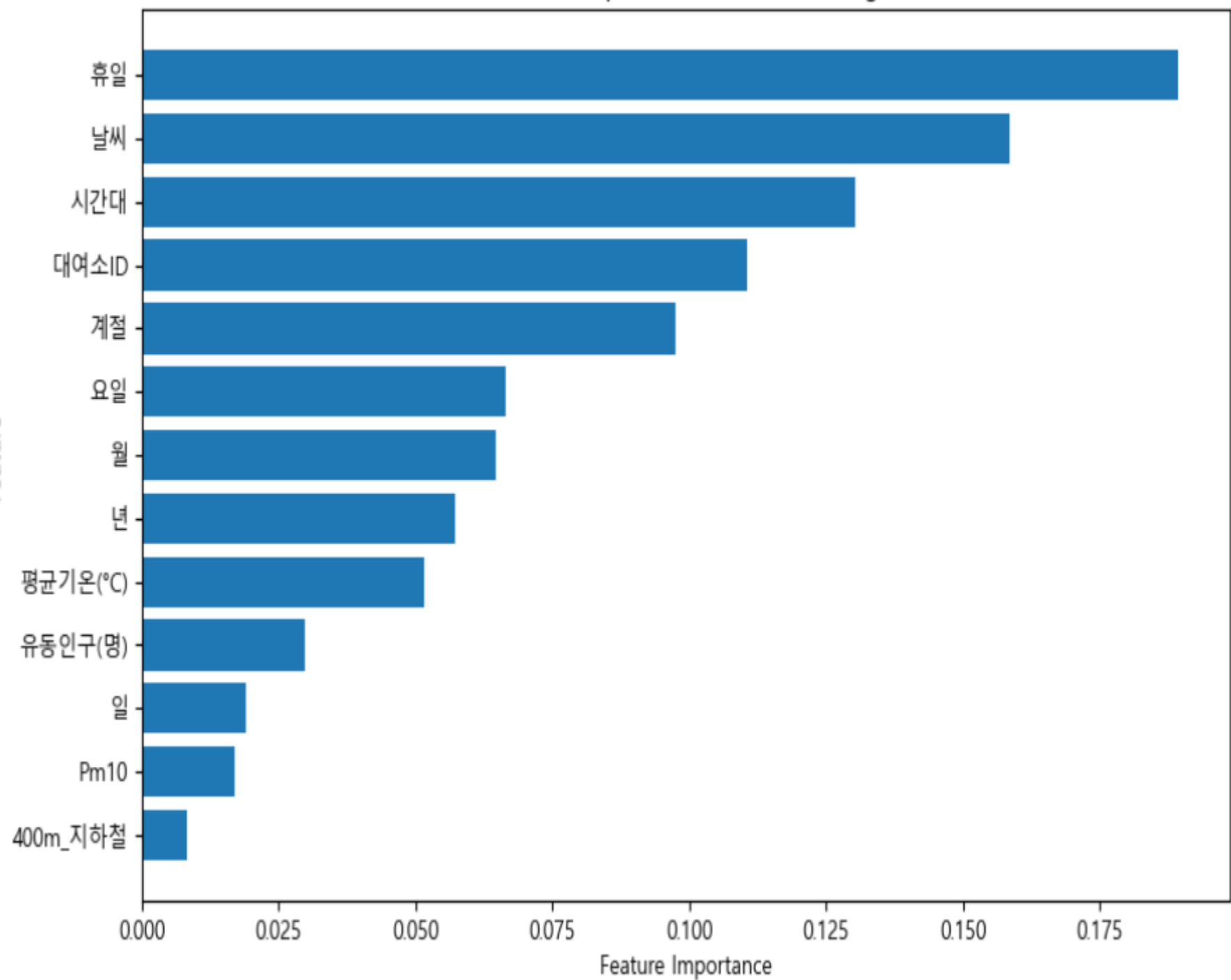
- 히트맵



피처중요도

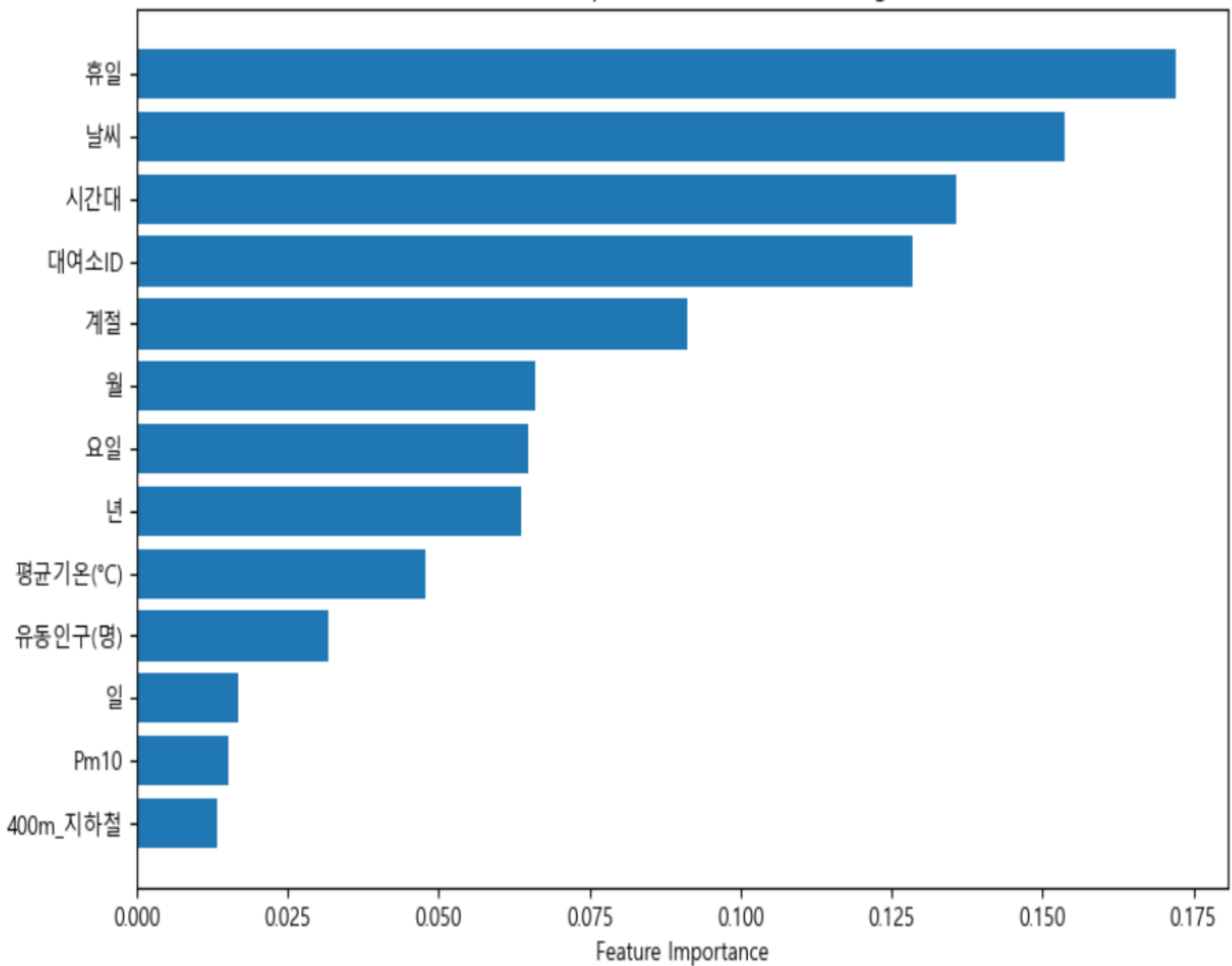


Feature Importance for Bike Renting



대여건수

Feature Importance for Bike Returning

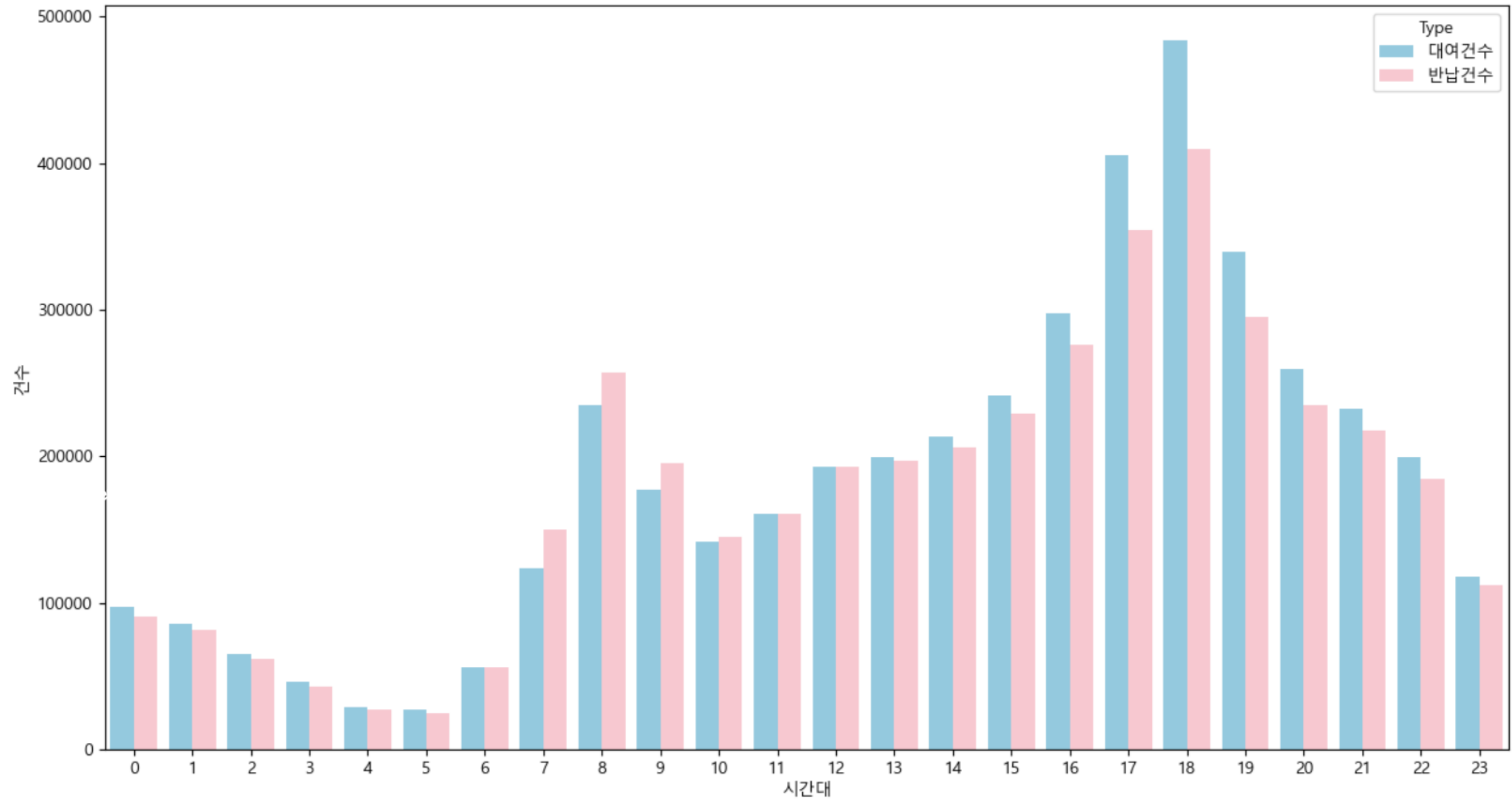


반납건수

시간대 추이



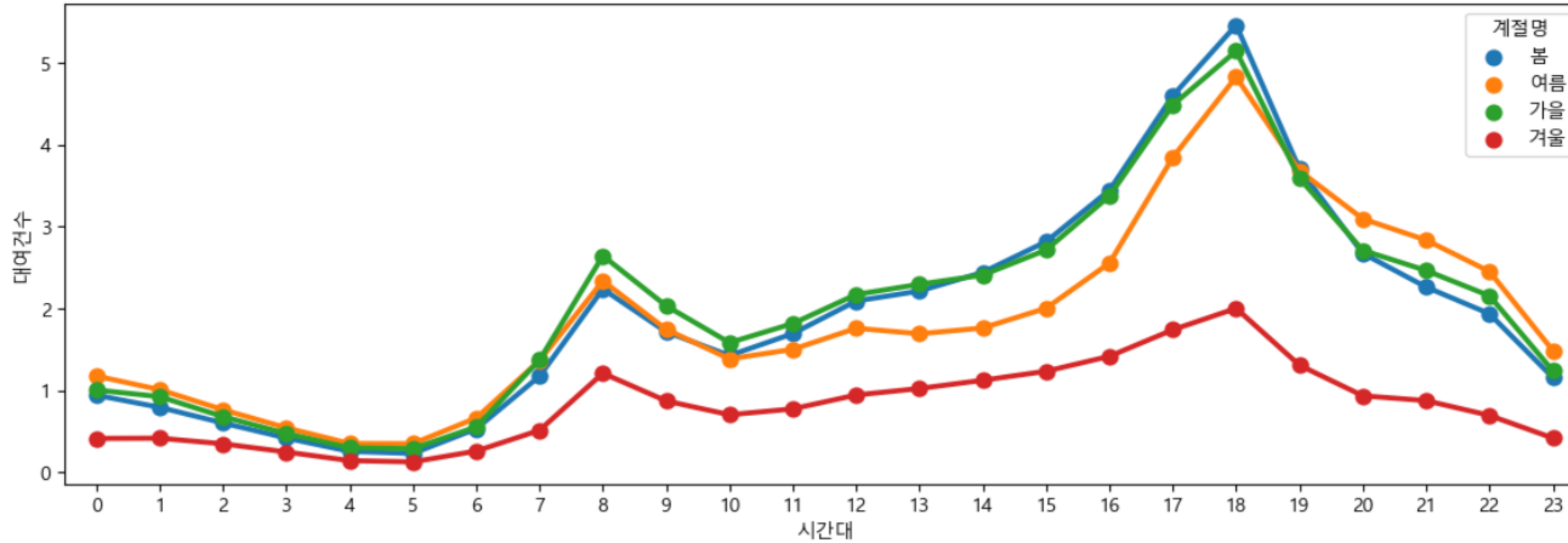
시간대별 대여건수 및 반납건수



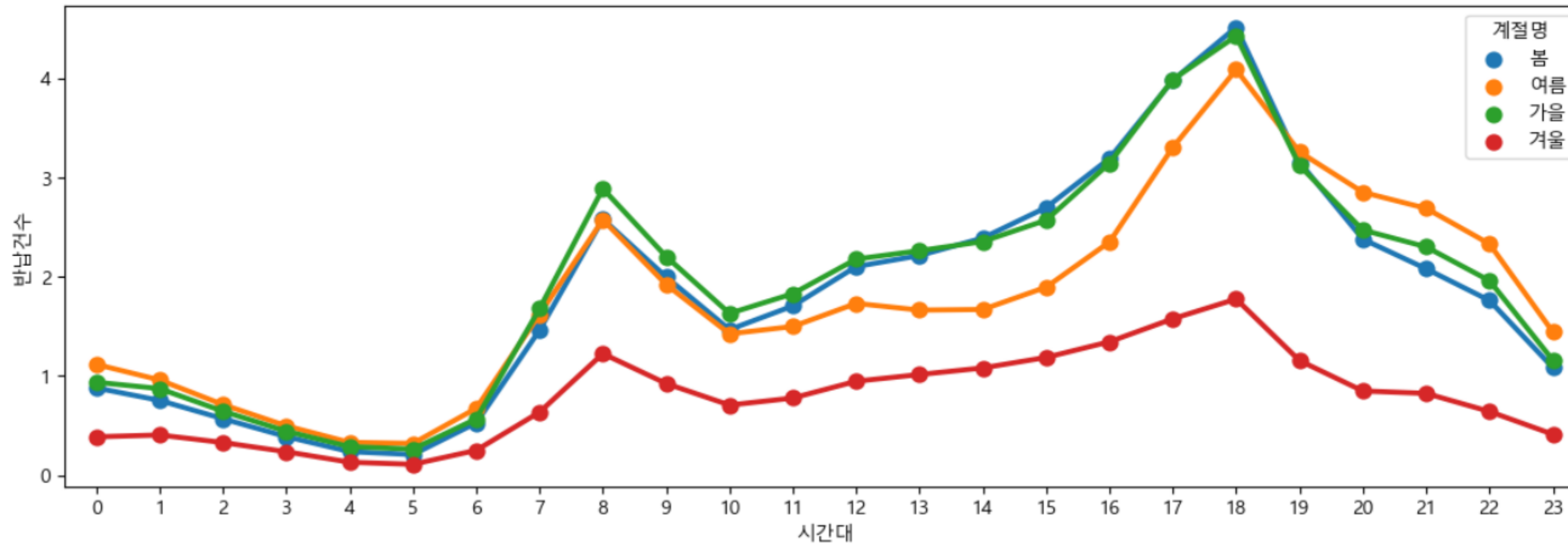
계절별 추이



계절에 따른 시간대별 평균 대여건수



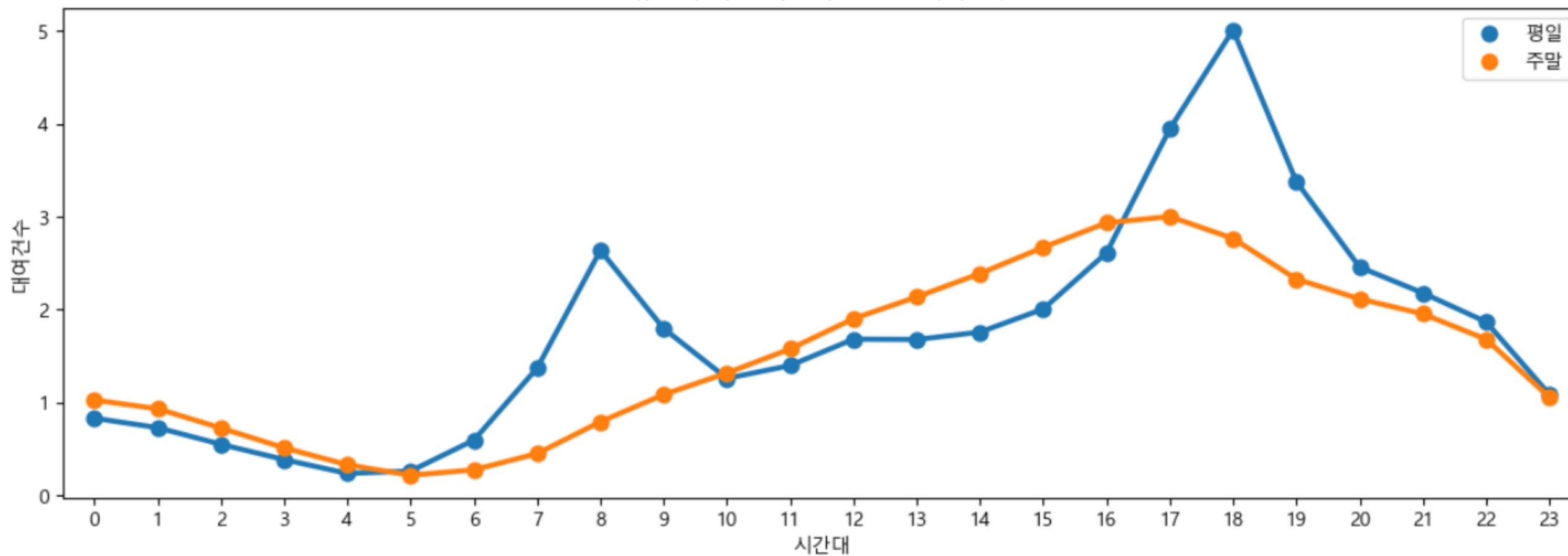
계절에 따른 시간대별 평균 반납건수



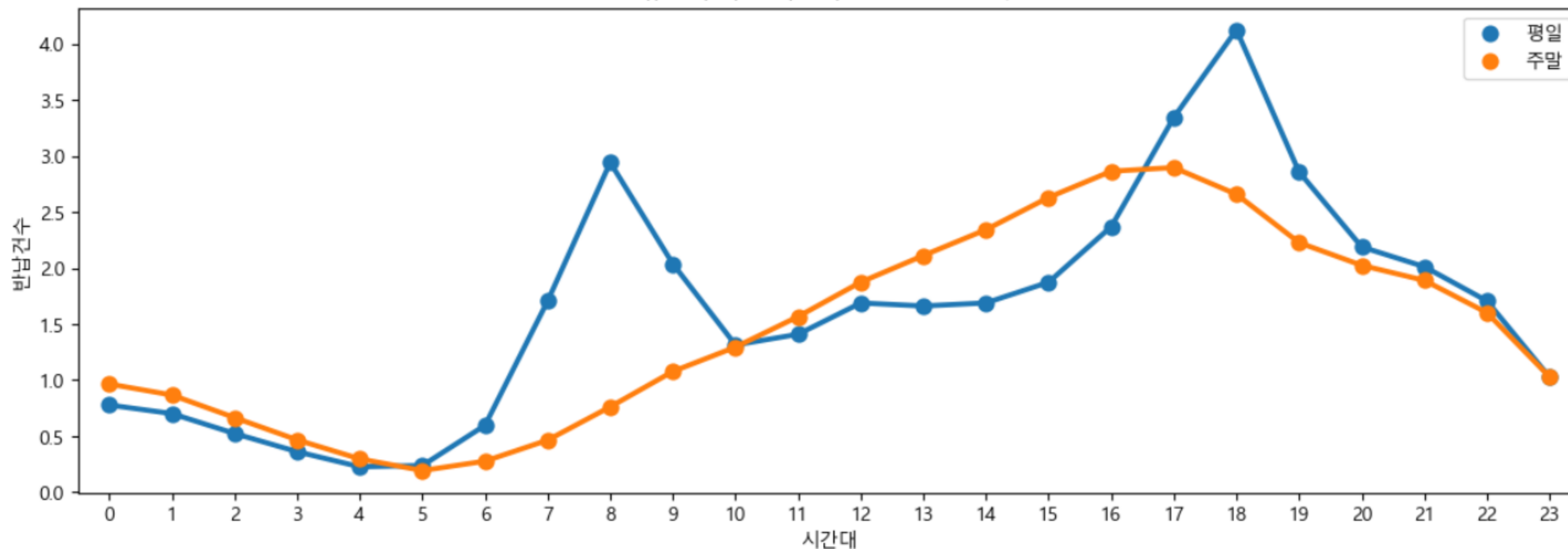
주간/주말 차이



휴일에 따른 시간대별 평균 대여건수



휴일에 따른 시간대별 평균 반납건수



Logistic Regression

(다중 클래스 분류)

정확도 : 23.91%

XGBoost

설명력 : 66.20%

MSE : 3.378 / MAE : 1.125

Linear(ols)

설명력 : 6.5%

MSE : 13.8349 / RMSE : 3.7195

Lasso

설명력 : 8.42%

RSMLE : 0.8070

Linear (Regression)

설명력 : 11.09%

MSE : 13.1820 / MAE : 2.2665

Ridge

설명력 : 8.40%

RMSLE : 0.806

RandomForest

설명력 60.3%

RMSLE : 0.5739

GradientBoostion

설명력 : 55.87%

RMSLE : 0.5935

LGBM

설명력 : 44.92%

MSE : 7.5036 / MAE : 1.8207

모델 결과



20~22년 존재하는 대여소 104개로 학습



✦ **대여량 설명력 : 61.11%**

✦ **반납량 설명력 : 66.20%**

MSE : 4.263, 3.378



LOWME



일반 위생

위생



시연 영상



컴파일러

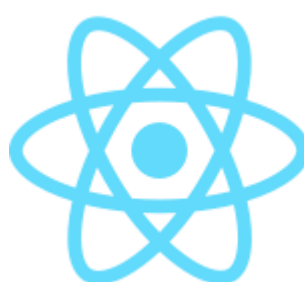


JAVA



파이썬

프론트



리액트



CSS3



AXIOS

백엔드



스프링부트



장고

라이브러리



시큐리티



웹소켓

DB



마리아DB



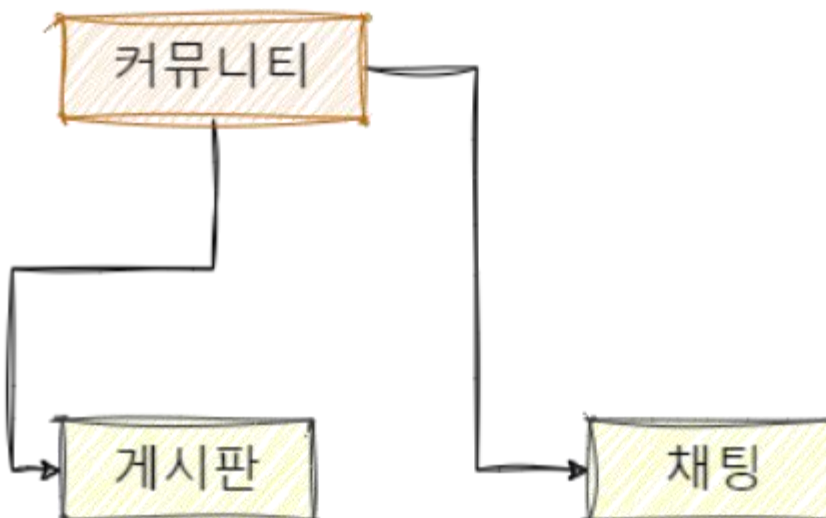
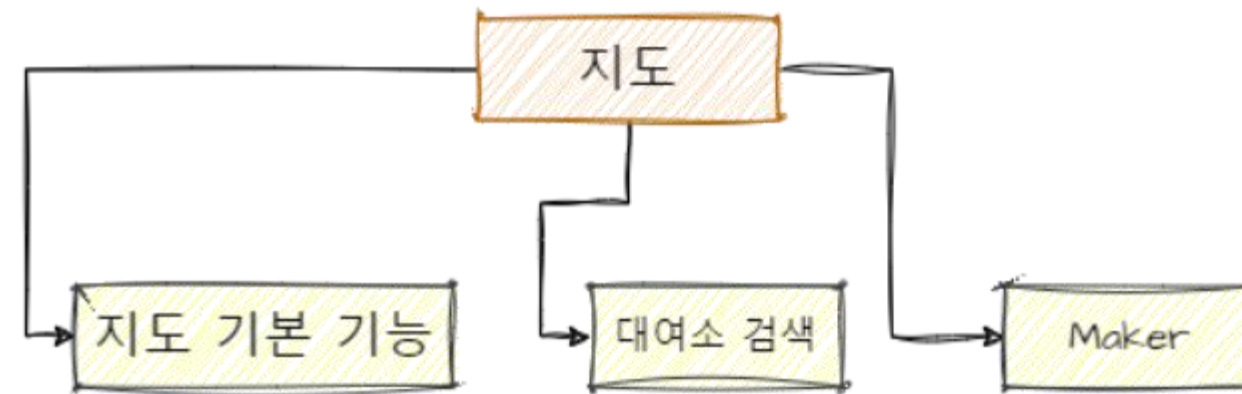
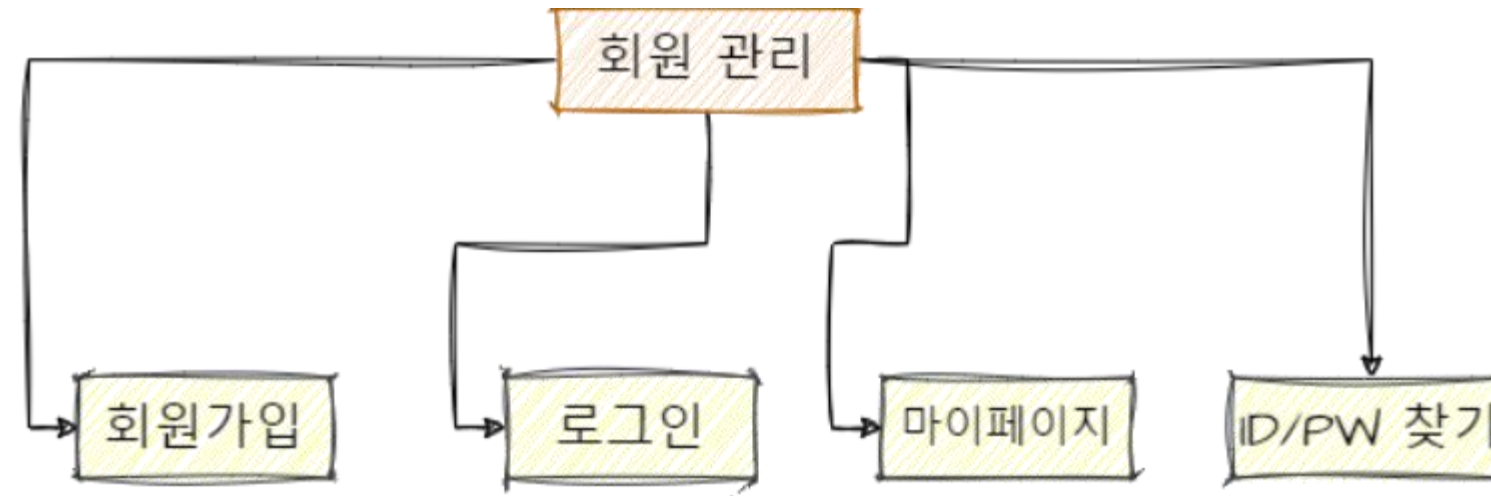
JPA

배포



도커

기능 소개



클라이언트

백엔드



지도 API



프론트엔드



Django

A X 1 O S

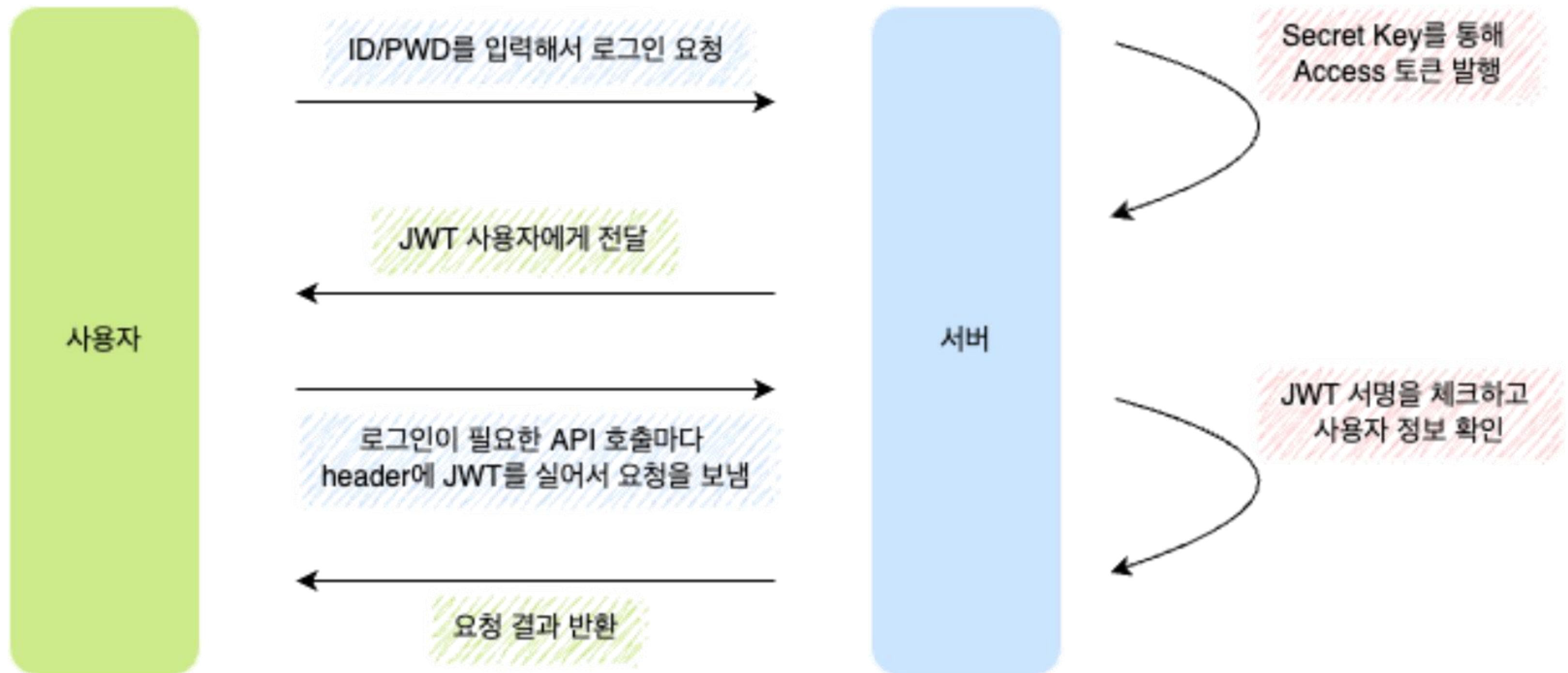
HTTP 통신 라이브러리

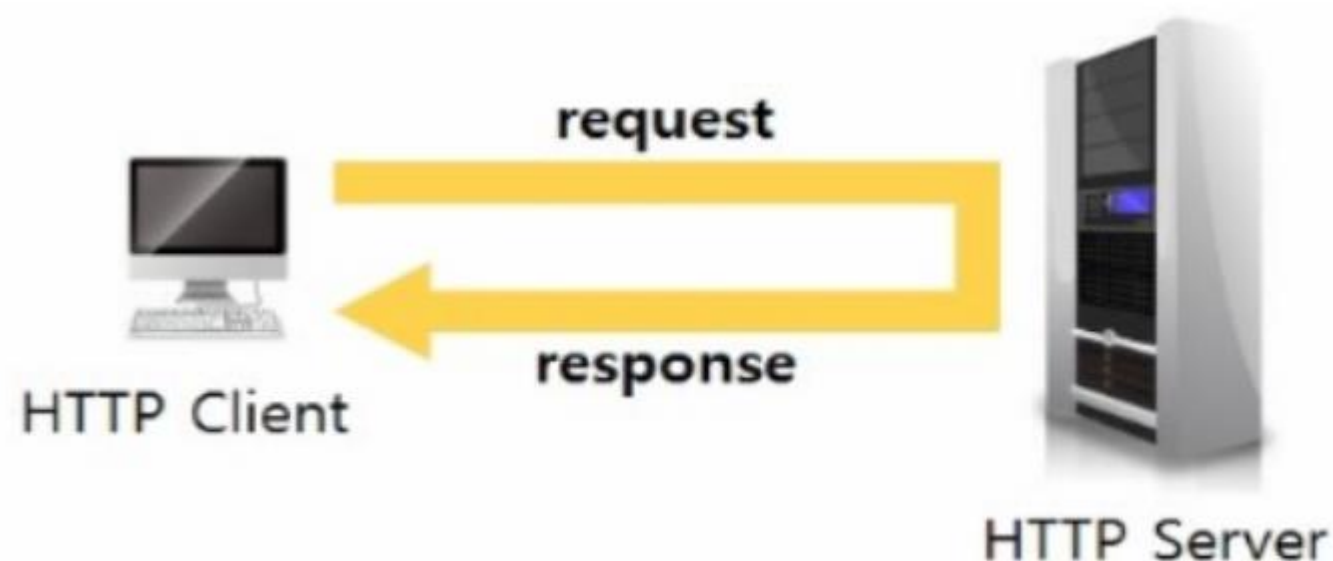
따릉이 대여소 목록 API를 요청하고 지도에 마커를 찍는다

Spring Security + JWT



Spring Security + JWT



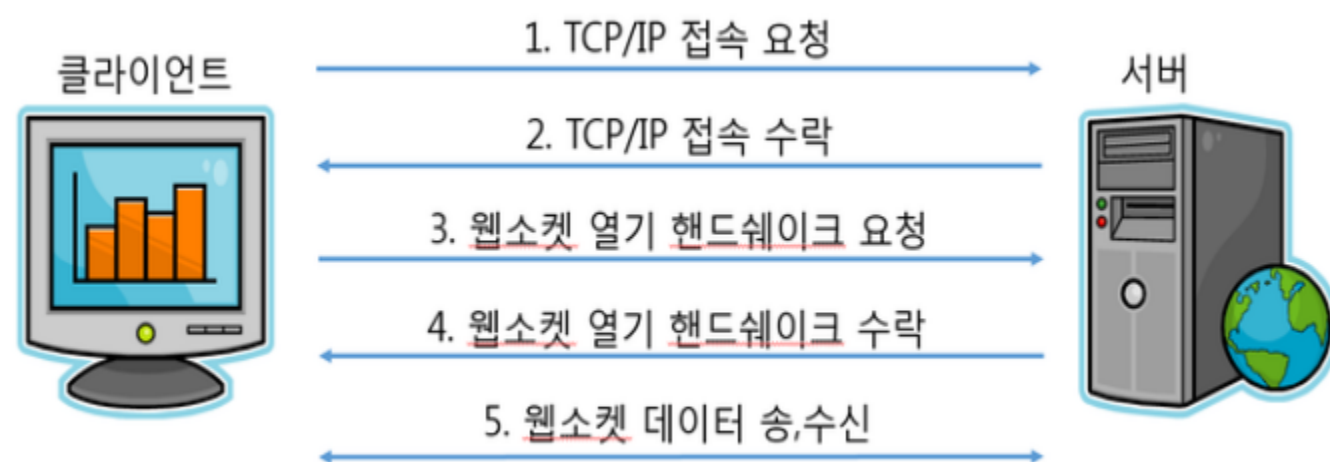


HTTP 통신과 웹 소켓의 차이

기존 HTTP 프로토콜 통신은 클라이언트가 요청을 보내고 서버가 응답을 반환한다. 서버는 클라이언트의 요청에만 반응하며, 클라이언트가 먼저 요청하지 않으면 서버는 자체적으로 통신을 시작할 수 없다.

하지만 웹소켓 방식은 클라이언트와 서버 간에 실시간 양방향 통신을 가능하게 하는 기술이다. 기존의 HTTP 프로토콜과 달리, 웹 소켓은 한 번의 연결 설정 후 계속해서 데이터를 주고 받을 수 있으며, 이로 인해 실시간 통신을 이용한 기능 구현에 적합하다.

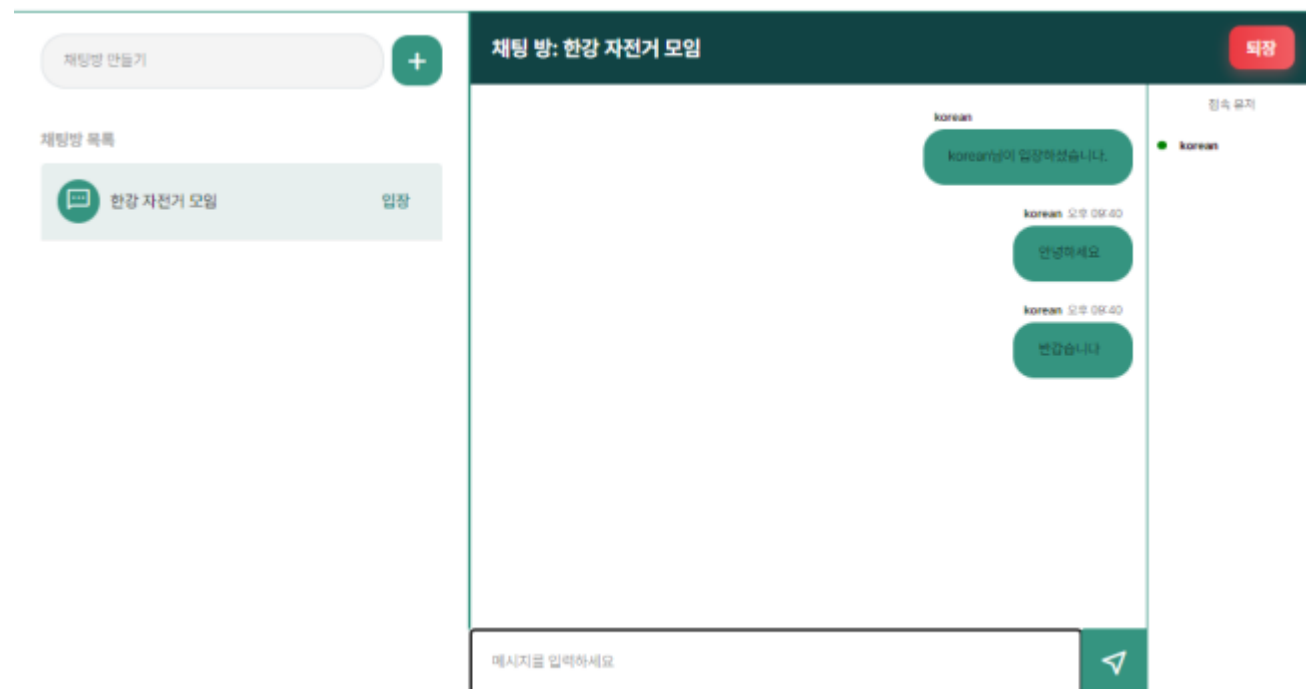
웹 소켓 접속 과정



웹소켓을 이용하여 서버와 클라이언트가 통신을 하려면 먼저 웹소켓 접속 과정을 거쳐야 한다. 웹소켓 핸드셰이크는 클라이언트가 먼저 핸드셰이크 요청을 보내고 이에 대한 응답을 서버가 클라이언트로 보내는 구조이다.

핸드셰이크는 웹 소켓 연결 설정 과정에서 클라이언트와 서버 간의 초기 교환 과정을 의미한다. 이 과정에서 서로의 존재를 확인하고, 필요한 프로토콜과 데이터 형식을 협상한다.

기능 소개



StompJS

StompJS는 웹소켓과 같이 사용하며 STOMP을 기반으로 하는 텍스트 지향 메시징 프로토콜이다. 웹소켓 위에서 작동하며 통신의 형식과 규칙을 제공하여 메시지 교환을 쉽게 만든다.

Topic, Sub, Pub의 의미

- Topic : 메시지를 발행하는 주제로, 구독자들은 특정 토픽을 구독하여 관련된 메시지를 받는다.
- Sub(Subscribe) : 특정 토픽이나 주제를 구독하는 행위를 의미한다.
- Pub(Publish) : 특정 토픽에 메시지를 발행하는 행위를 의미한다.

전체적인 Stomp의 흐름

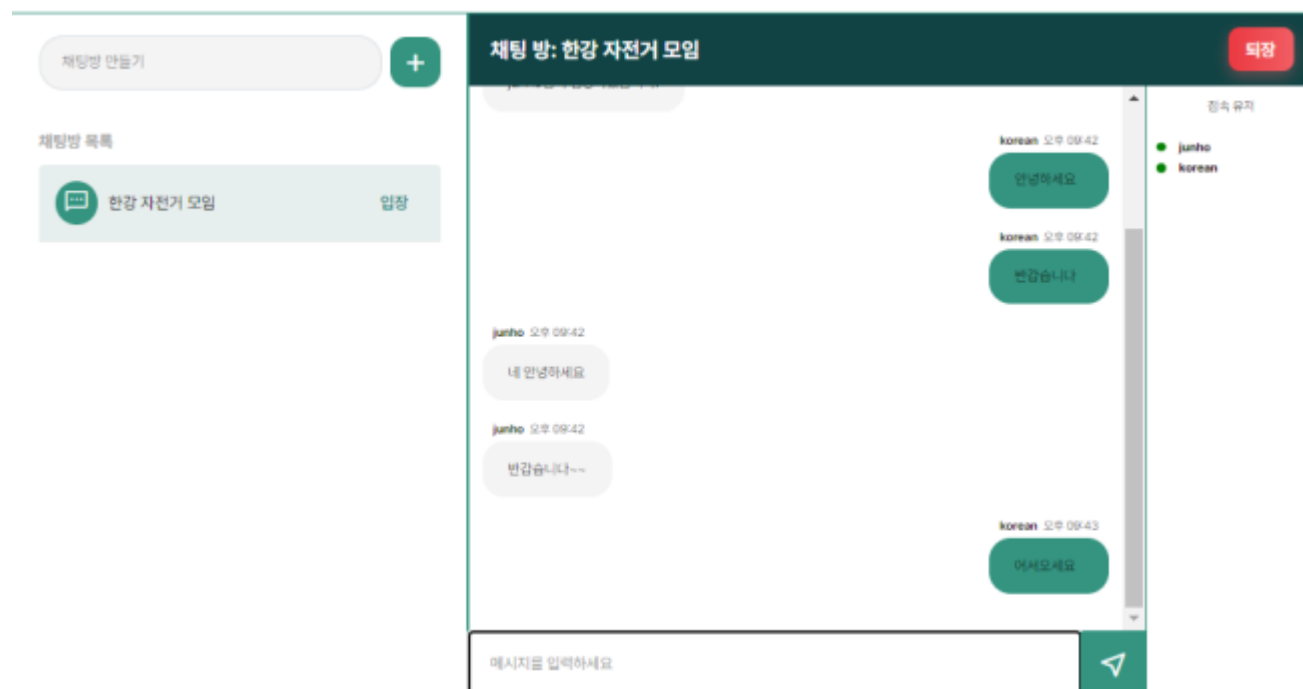
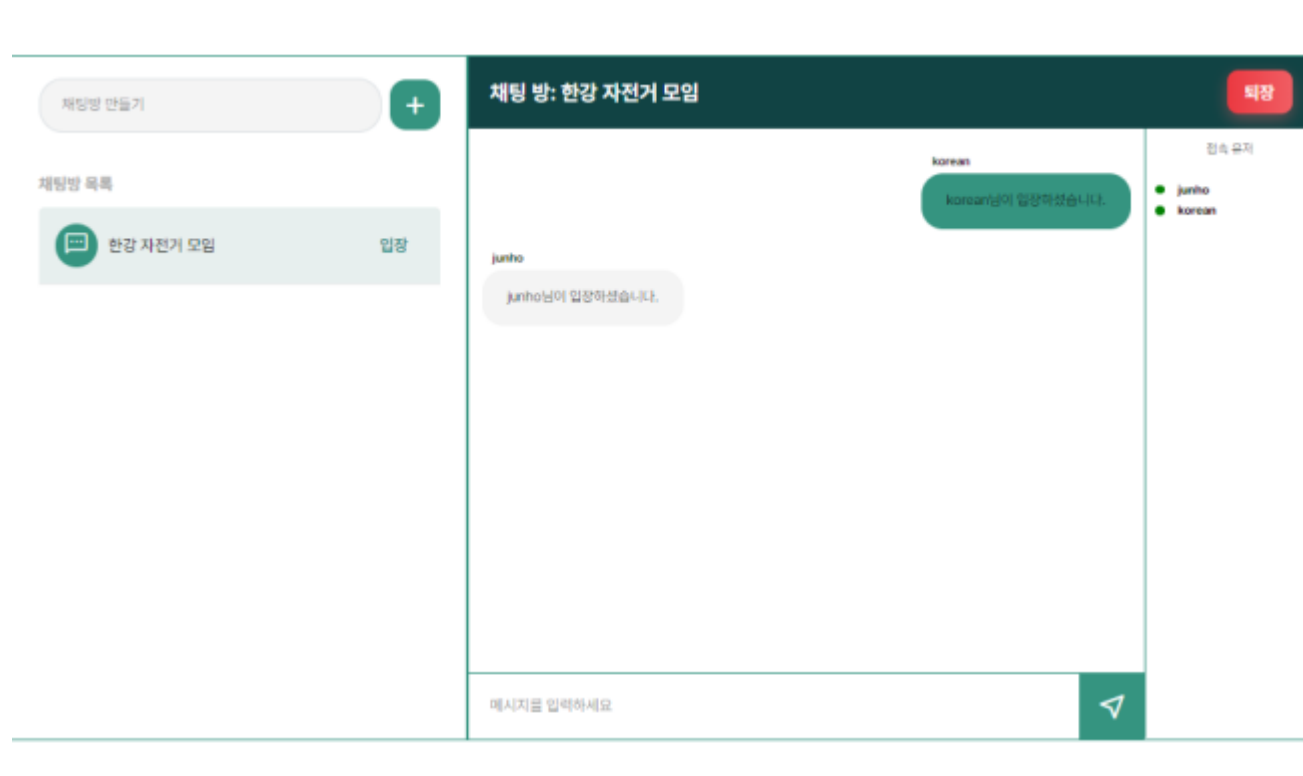
중개 서버를 통해서 클라이언트간에 비동기적으로 메시지를 전송하기 위한 프로토콜

- 서버와 연결할 클라이언트 Connection
- 메시지 전송 전 Subscriber와 Publisher를 지정
- Subscribe를 하면 해당 URL로 나에게 메시지를 보낼 수 있는 경로가 생긴다
- Publisher를 하면 Publish한 URL로 메시지가 이동한다

```
0개의 사용위치  gukin
@Override
public void configureMessageBroker(MessageBrokerRegistry config) {
    config.enableSimpleBroker(...destinationPrefixes: "/sub");
    config.setApplicationDestinationPrefixes("/pub");
}

// WebSocket 서버 주소 및 포트 설정
const socket : SockJS = new SockJS( url: `http://localhost:8080/ws-stomp?access_token=${encodedToken}`);
// STOMP 클라이언트 생성 및 연결 설정
stompClient.current = Stomp.over(socket);
stompClient.current.connect({}, (frame) : void => {
    setConnected( value: true);
    subscribeToRoom(roomId);
});
```


기능 소개



전체적인 작동 흐름

사용자가 채팅방에 입장하면 리액트 컴포넌트에서 SockJS와 Stomp를 이용해 웹소켓 연결을 시도하고, 3가지 타입의 메시지를 서버로 보낸다.

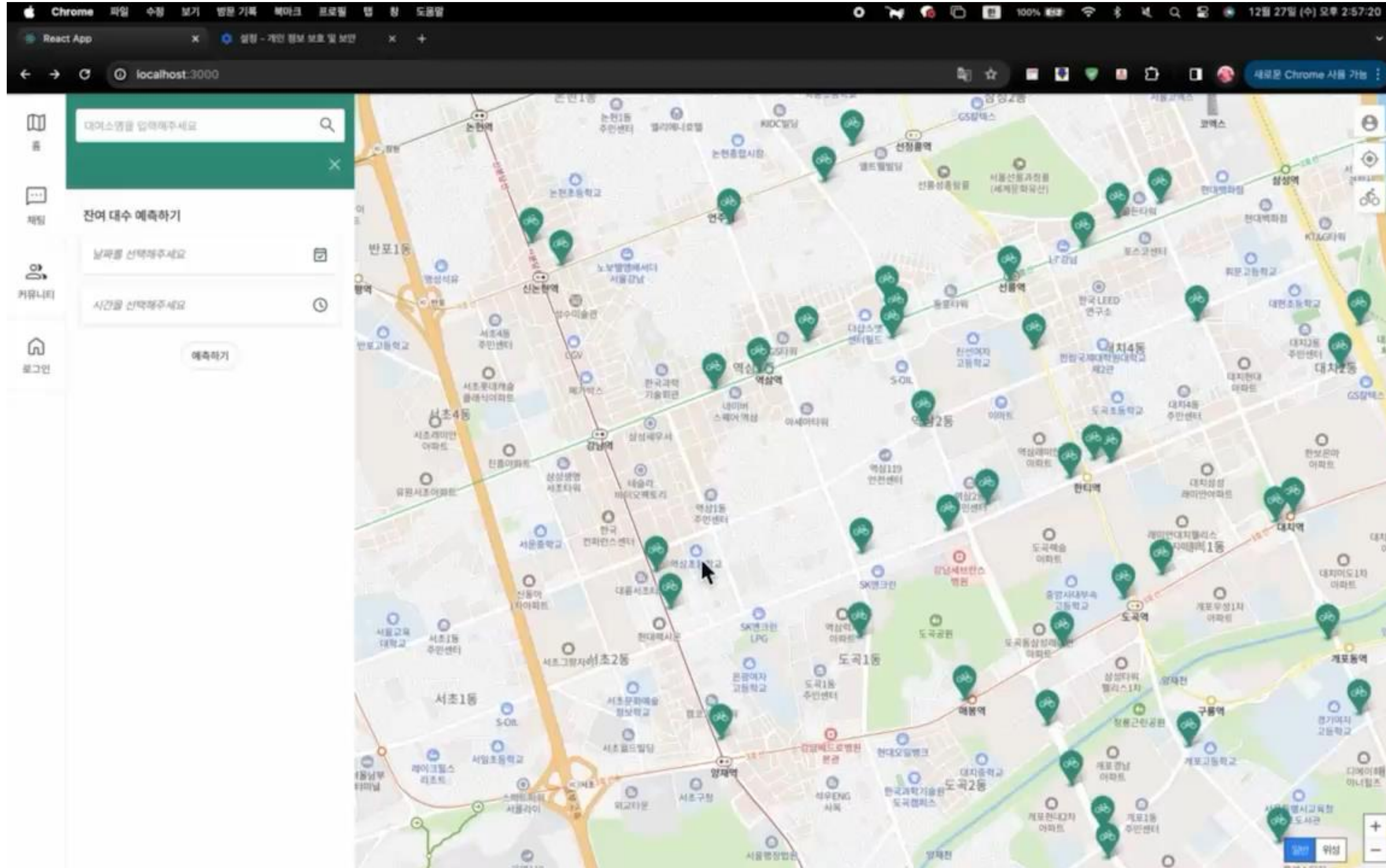
1. ENTER : 입장 메시지를 받고, 유저 목록을 업데이트하며 해당 정보를 다시 채팅방 구독자들에게 보냄
2. TALK : 사용자가 메시지를 보낼 때, 사용되는 유형
3. EXIT : 사용자가 채팅방에서 나가면 EXIT 메시지가 서버로 전송되고, 유저 목록이 업데이트

웹소켓 연결이 설정되면 사용자는 메시지를 입력, 전송이 가능하며 메시지는 JSON 형식으로 서버로 전송
서버에 새로운 메시지가 도착하면 웹소켓을 통해 메시지를 데이터베이스에 저장 후 채팅방에 표시

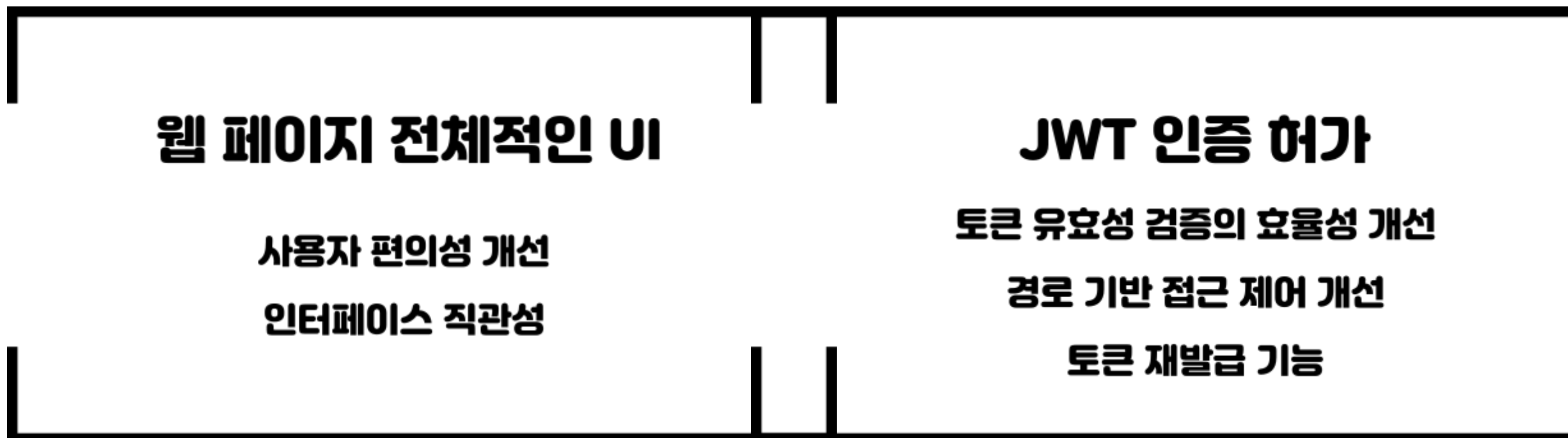
JWT를 이용한 웹 소켓 연결 후 통신

1. WebSocket 연결 : 클라이언트가 WebSocket을 사용하여 서버에 연결을 시도. 액세스 토큰을 이용
2. 토큰 전송 : 클라이언트는 URL 파라미터로 액세스 토큰을 전송. 핸드셰이크 과정에서 서버로 전달
3. 서버에서 토큰 검증 : WebSocket 서버에서는 핸드셰이크 인터셉터를 사용하여 핸드셰이크 과정에서 클라이언트가 전송한 토큰을 검증
4. 토큰 디코딩 : 서버에서 토큰을 검증한 후, 토큰에 포함된 정보를 디코딩하여 아이디를 추출
5. 세션 정보 저장 : 추출한 사용자 아이디를 WebSocket 세션 ID와 매핑을 유지하고, 사용자를 연결
6. WebSocket 통신 : WebSocket 연결 후, 클라이언트는 WebSocket 연결을 통해 사용자 아이디와 관련된 데이터를 서버로 전송하고, 서버는 이 데이터를 활용하여 처리

시연 영상



보완점



허준 매일 매일 회의와 소통을 통해 하나의 결과물을 만들어 내는 프로젝트 경험이 정말 유익하고 의미 있는 시간이었습니다. 이 경험을 통해 저희 팀은 협력과 목표 달성을 위한 열정적인 노력을 보여주었으며, 결과물을 만들어내기 위한 힘을 모아 나갈 수 있었습니다.	김예 다양한 기술을 적용하고 팀원들과의 협업 경험을 키울 수 있어서 매우 좋은 시간이었습니다. 어려운 순간들도 있었지만, 그로 인해 팀 협력과 문제 해결 능력을 키울 수 있는 소중한 기회가 되었습니다.	한국 두 번의 프로젝트를 통해 여러 기술들을 쓰며 다양한 경험을 할 수 있어 좋았습니다. 중간 프로젝트에 이어 최종 프로젝트까지 함께 고생한 팀원들한테 고맙고 고생한만큼 다들 앞으로 좋은 결과만 있었으면 좋겠습니다.
김민 수백번의 회의와 소통을 통해 나오게 된 결과물입니다. 데이터탐색부터 전처리, 모델생성 등 어느 하나 쉬운일이 없었습니다. 하지만 끝내 해냈고 그렇기에 아쉬움보단 뿌듯함이 더 큼니다. 모두 좋은 곳에 취업하길 바라며 모두 정말 감사했습니다!	안준 저번 프로젝트와는 달리 팀원들과 얘기를 소통을 많이함으로써 원활하게 진행할 수 있었습니다. 물론, 프로젝트를 진행하면서 어려움도 존재하였지만 함께 하는 팀원들이 있어 재미있었고, 그래서 더더욱 끝까지 포기하지 않고 마칠수 있었던 것 같습니다.	윤정 데이터분석의 전반적인 프로세스를 경험할 수 있어서 좋은 경험이 되었습니다. 중간 프로젝트의 경험을 토대로 아쉬웠던 점은 보충했습니다. 프로젝트를 통해 더욱 성장할 수 있는 계기가 되었습니다.

자유롭게 질문해주시기 바랍니다

ACADEMY@acorn.com



00-123-4567



서울특별시 강남구 역삼동
삼원빌딩 5층



감사합니다