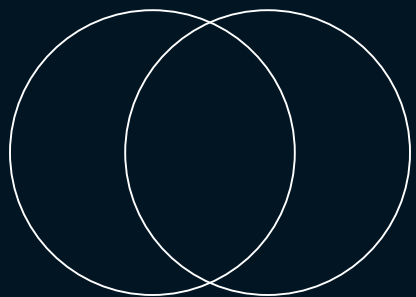


# 부동산 가격 예측 모델 개발

팀 이름: 아파트

조 원: 이유리, 우병준, 이정인



# Table of Contents

<b>1. Project Overview</b>	프로젝트 개요
<b>2. Team Composition and Roles</b>	팀 구성 및 역할
<b>3. Project Execution Process and Methodology</b>	수행 절차 및 방법
<b>4. Data Preprocessing</b>	데이터 전처리
<b>5. Project Outcomes - Modeling</b>	수행결과 - 모델링
<b>6. Self-Evaluation and Insights</b>	자체 평가 의견

# 매매가 예측

## 주제 선정 배경

서울시 아파트 매매가는 부동산 시장의 주요 지표. 따라서 시민과 투자자들에게 중요한 의사결정 정보를 제공

### 프로젝트 개요

서울시 아파트 매매가 예측 머신러닝 모델 개발

주요 데이터  
아파트 매매 실거래가 데이터 등

데이터 전처리 및 분석  
feature 선정  
예측 모델 구성

### 활용 장비 및 개발 환경

언어  
Python

라이브러리  
Pandas, NumPy, Scikit-learn, XgBoost 등

분석 도구  
Jupyter Notebook, VS Code

### 프로젝트 구조

데이터 수집

데이터 전처리 및  
탐색적 분석

머신러닝 모델  
구축 및 예측

## 2. Team Composition and Roles

## 팀 구성 및 역할

이유리

우병준

이정인

수집  
데이터

- 아파트 실거래가
- 국토교통부
- **아파트 브랜드**
- **부동산 사이트**
- 구/행정동/법정동
- 공공데이터 포털 및 브이 공간 월드
- 버스정류장, 지하철역
- 공공데이터 포털
- **정치 데이터**
- **재임 기간**

- 소비자 물가지수
- 통계청
- **환율, 금리**
- **한국은행**
- 행정동별 인구밀도
- 공공데이터 포털
- **행정동별 상가 소득 및 소비**
- **공공데이터 포털**
- 범죄율(폐기)

- 학교, 병원, 공원
- 공공데이터 포털 위치 데이터
- 날씨(폐기)
- 기상청

전처리

- 그 외 모든 전처리

- 2019년, 2020년 아파트 도로명 주소 위도 경도 변환
- 행정동별 상가 소득 및 소비

- 학교 도로명 주소 위도 경도 변환

모델링

- XgBoost
- Ridge, Lasso
- Decision Tree

- Random Forest
- 최고 성능 모델의 하이퍼파라미터 튜닝

기타

- 발표자료 제작 (전반부)
- 문서 작업

- 발표자료 제작 (후반부)
- 문서 작업

3. Project Execution Process and Methodology

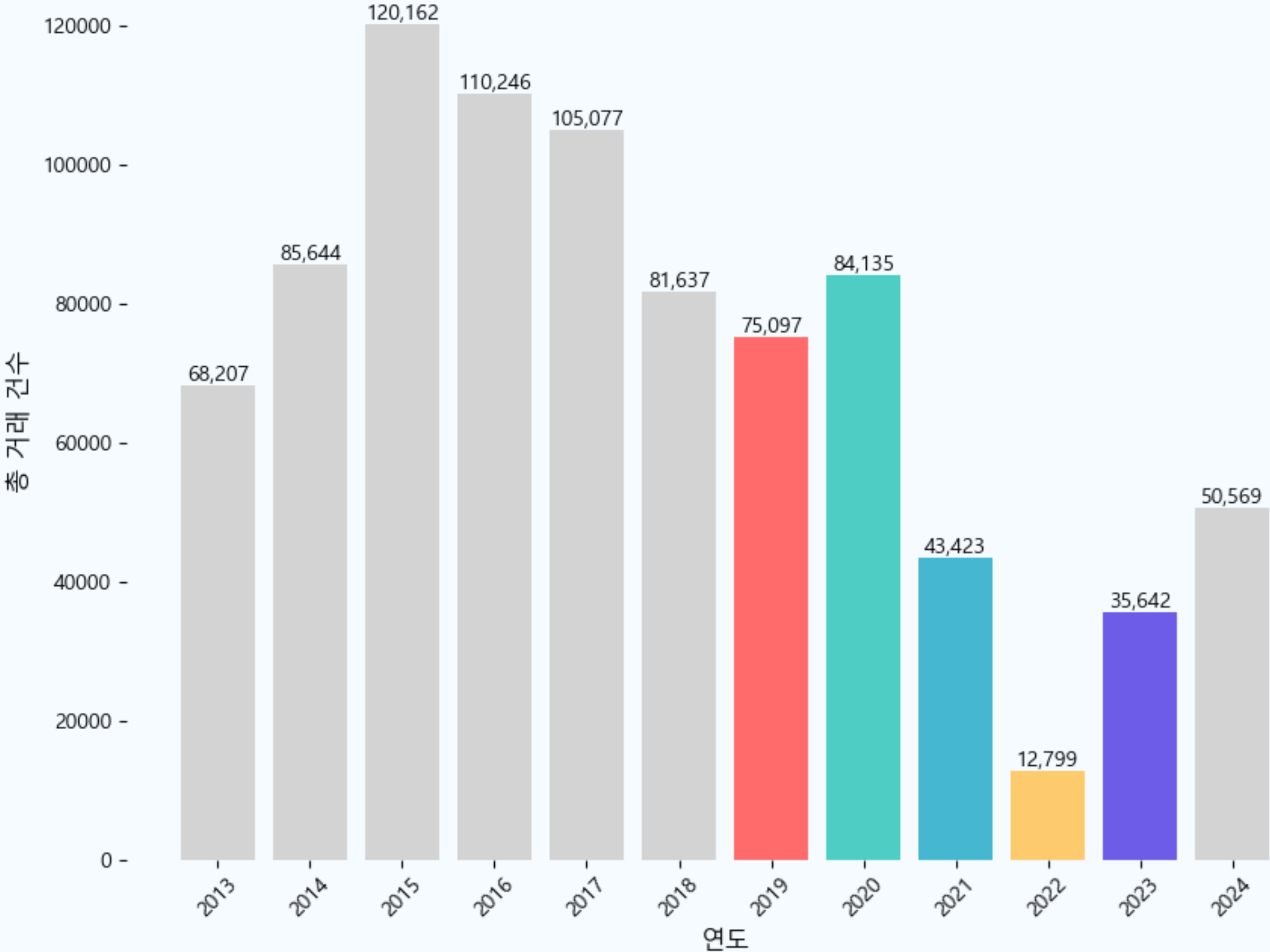
수행 절차 및 방법

기획	수행		
프로젝트 주제 선정	데이터 수집	데이터 전처리	예측 모델링
<div><ul style="list-style-type: none"><li>부동산 시장 분석 및 프로젝트 주제 선정</li><li>필요 데이터 및 목표 정의</li></ul></div>	<div><ul style="list-style-type: none"><li>서울시 실거래가 데이터 수집</li><li>주변 시설</li><li>교통</li><li>정치</li><li>환율, 금리</li><li>소득-소비</li><li>브랜드화</li></ul></div>	<div><ul style="list-style-type: none"><li>아파트 기본 정보 주소 위도경도 변환 등</li><li>주변 시설 및 교통 아파트 기준 최단거리 계산 일정 반경 내 점수화</li><li>동별 직전 월의 거래 건수</li><li>환율, 금리, 정치 결측치 처리</li><li>인구밀도 및 경제 수준</li></ul></div>	<div><ul style="list-style-type: none"><li>Ridge, Lasso</li><li>Decision Tree</li><li>Random Forest</li><li>XgBoost</li><li>최고 성능 모델 (Gradient Boosting 알고리즘 구현) XGBoost 하이퍼파라미터 튜닝</li></ul></div>

# 4. Data Preprocessing

## 데이터 전처리

연도별 총 거래 건수 (2019-2023 하이라이트)



2019년 75,097건

2020년 84,135건

2021년 43,423건

2022년 12,799건

2023년 35,642건

5개년 총합 251,096건

## 4. Data Preprocessing

## 데이터 전처리

불필요한 컬럼을 지운 서울시 아파트 실거래가 원데이터 (도로명 주소는 가림)

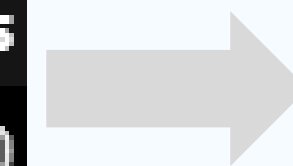
NO	시군구	단지명	전용면적 (㎡)	계약년 월	계약 일	거래금액(만 원)	층	건축년 도
1	서울특별시 성동구 마장 동	현대	134.790	201912	31	88,000	17	1998

인덱스용  
NO 컬럼 값 정제

NO
2019_00001



NO	address
2019_00001	서울특별시 성동구 살 [가림] 0



NO	latitude	longitude
2019_00001	37.56 [가림]	127.04 [가림]

위도 경도 추출을 위  
한 주소 정제

API 활용  
위도 경도 변환

Missing Value는 주소를 뒤에서 부터 잘라,  
**모든 주소 위도 경도 변환**  
(학교 주소도 같은 방식으로 변환)

# 4. Data Preprocessing

## 데이터 전처리



상위 1위 ~10위  
고급 (4점)

상위 11위~28위  
중급 (3점)

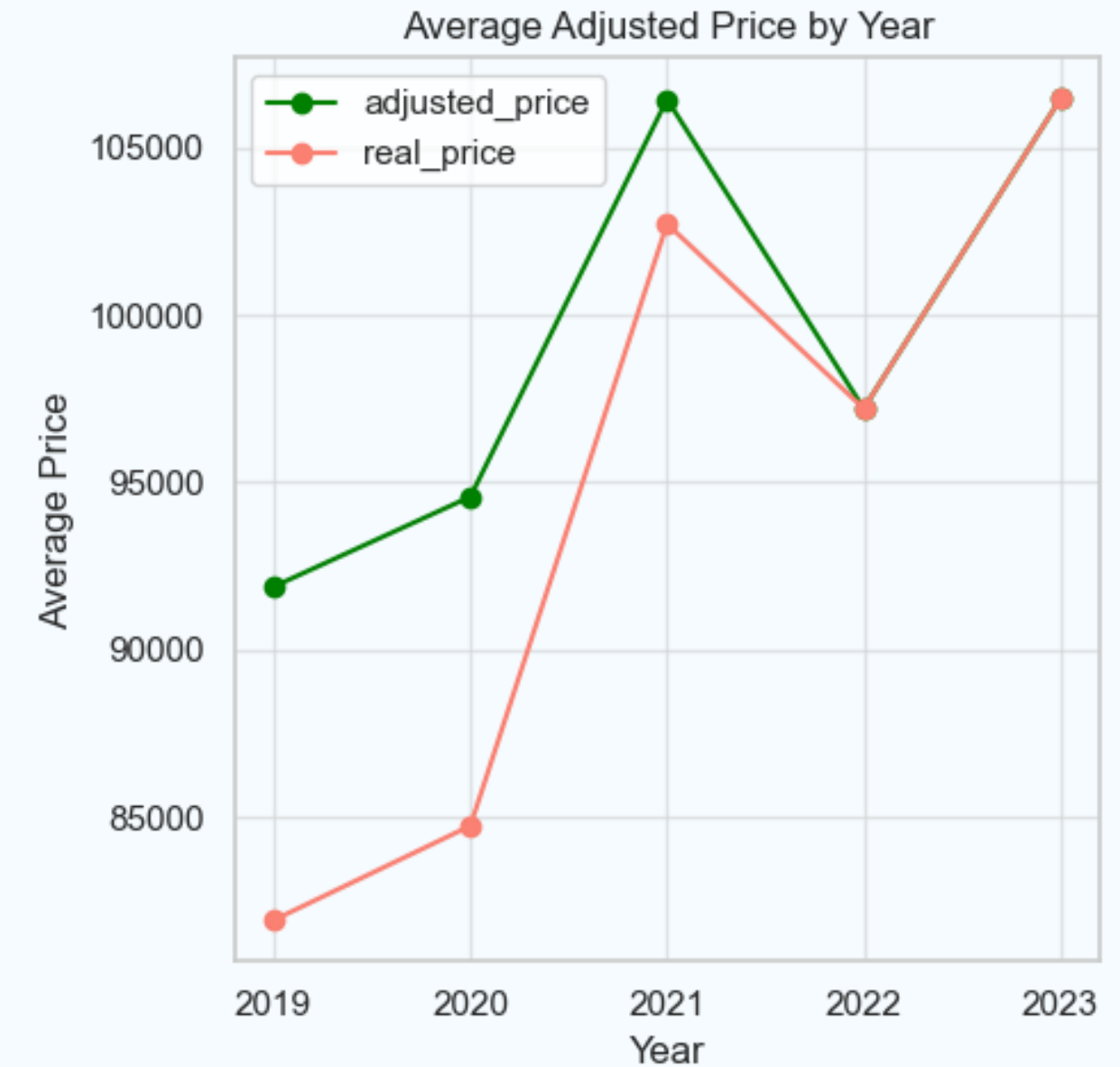
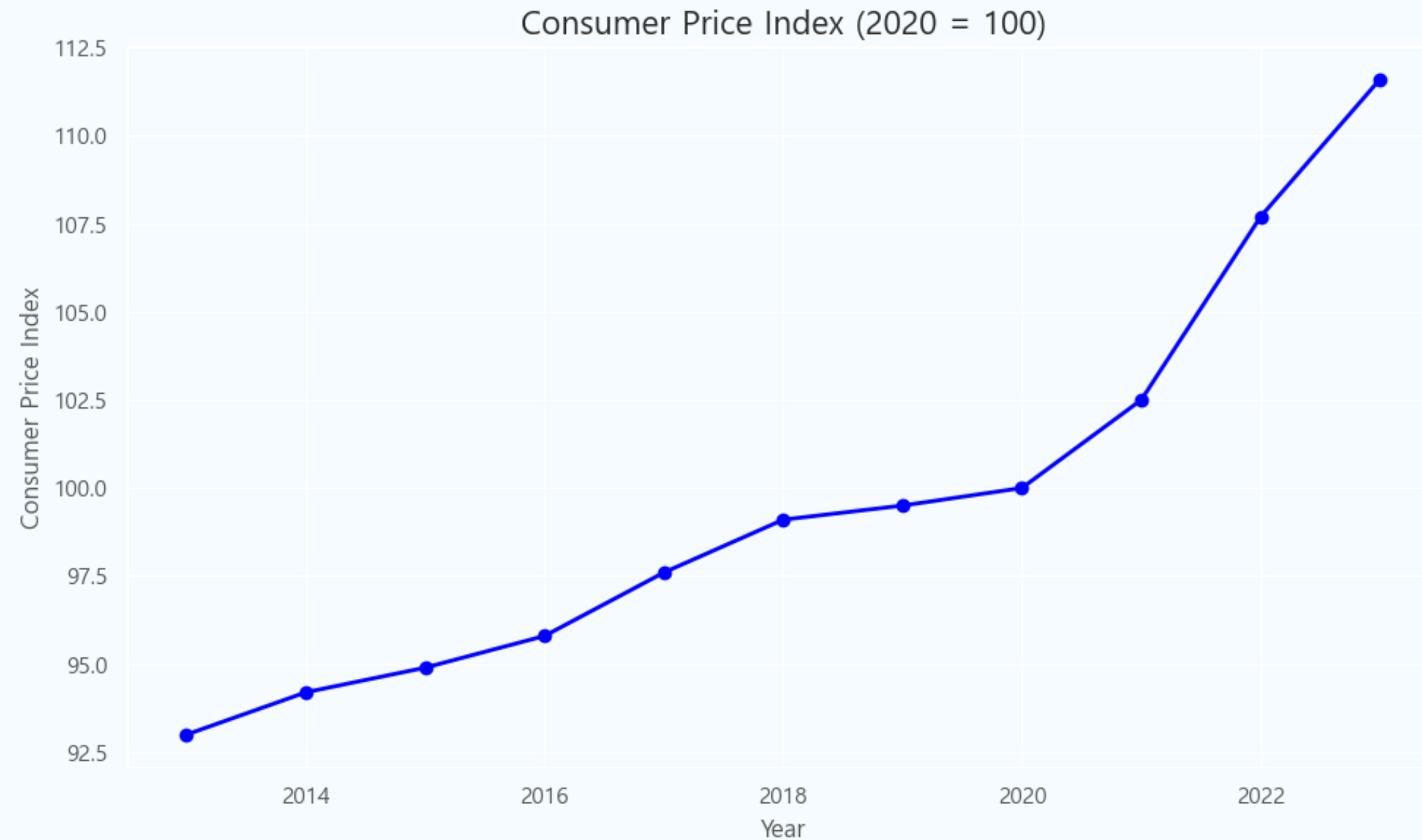
순위 밖 브랜드  
하급 (2점)

그 외  
브랜드 아님 (1점)

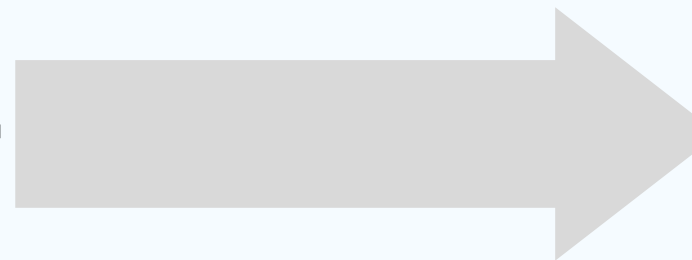


## 4. Data Preprocessing

## 데이터 전처리



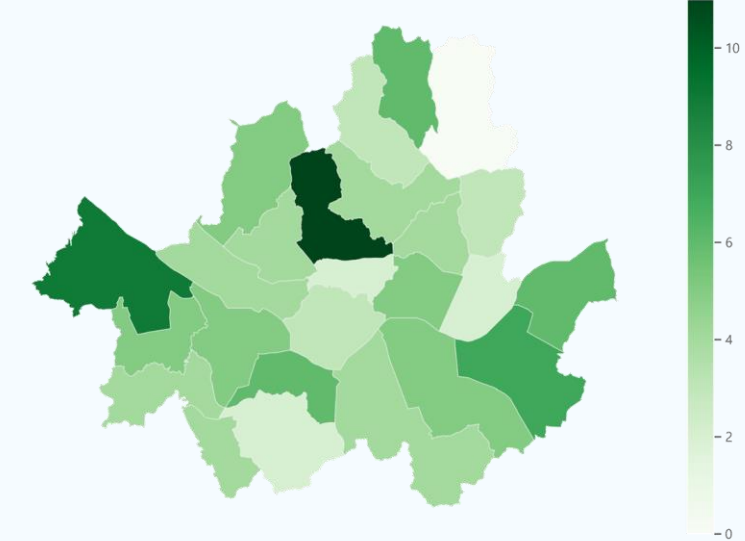
소비자 총물가지수 활용



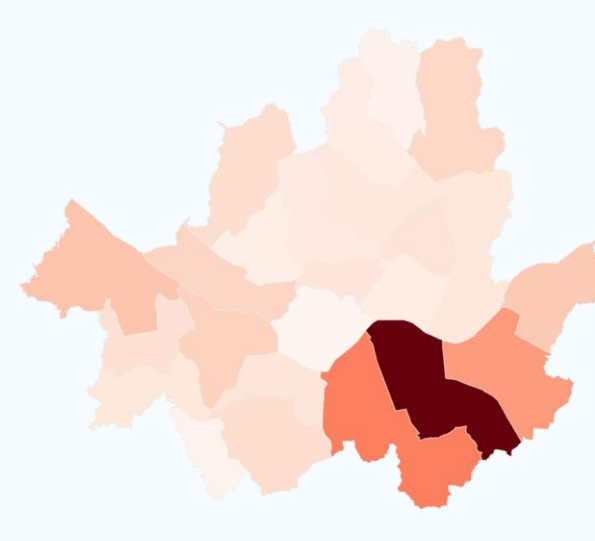
2023 화폐가치로 전환

## 4. Data Preprocessing

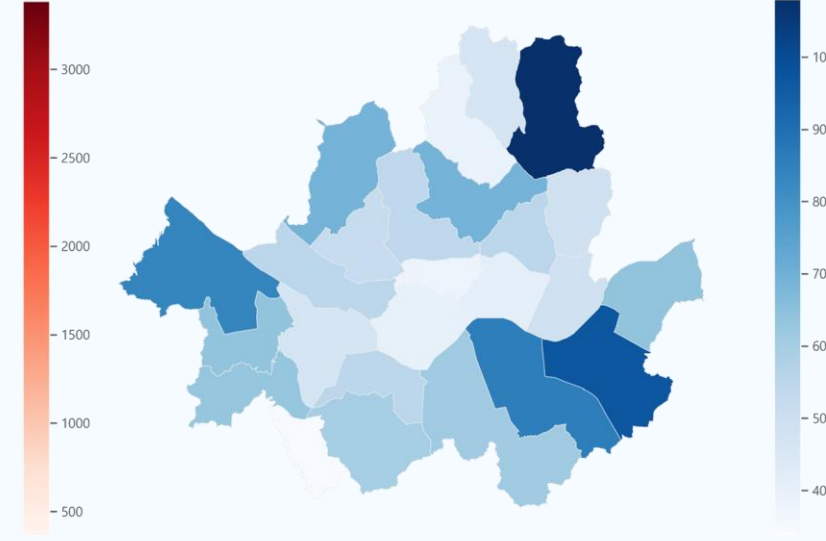
서울시 구별 공원 개수



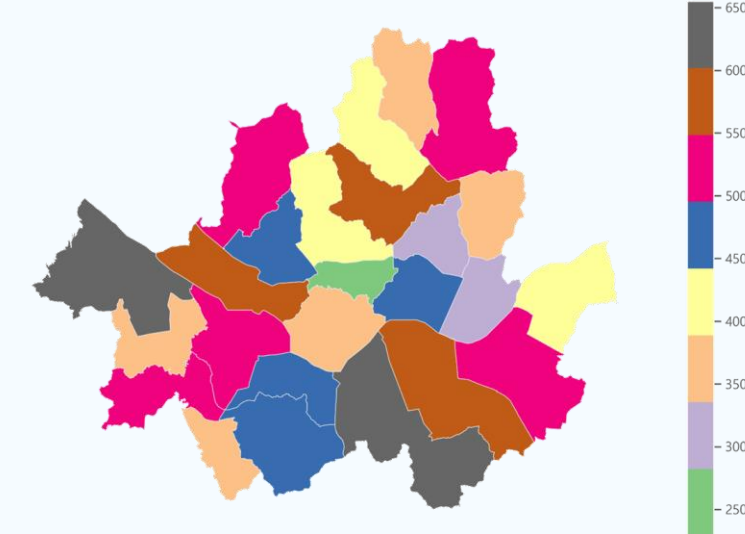
서울시 구별 병원 개수



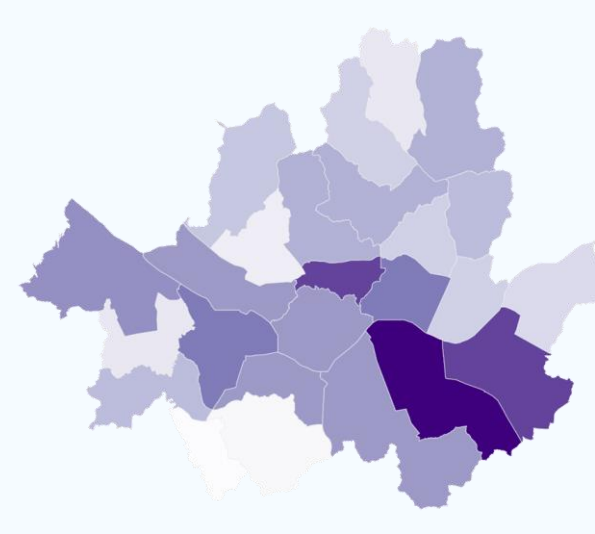
서울시 구별 학교 개수



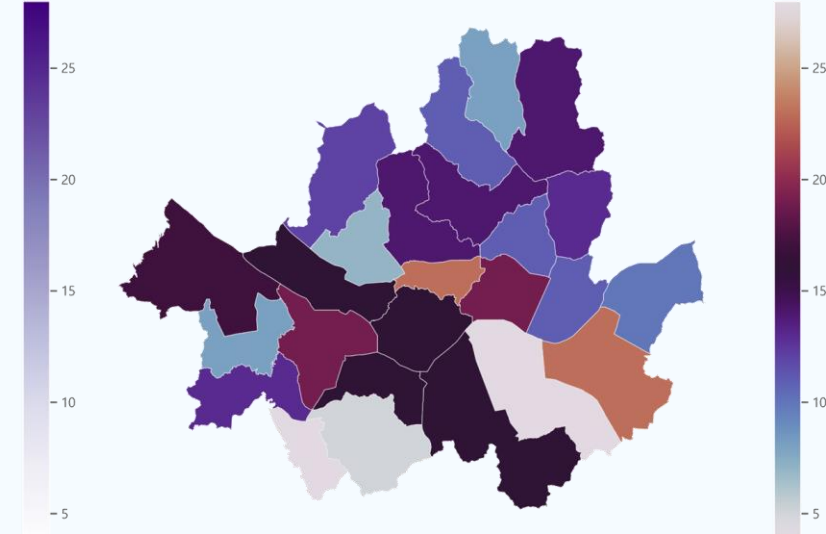
서울시 구별 버스정류장 개수



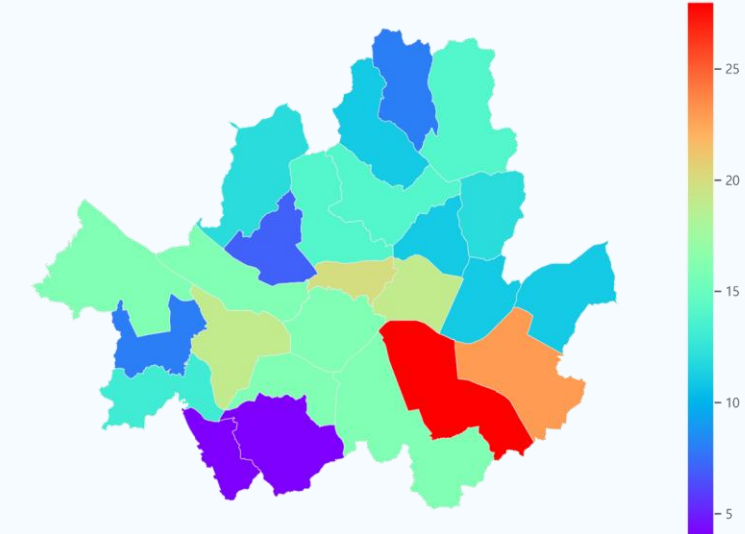
서울시 구별 2019년 지하철역 개수



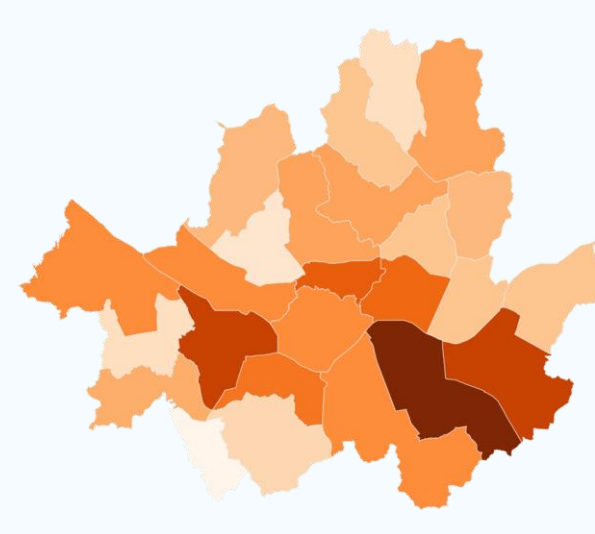
서울시 구별 2020년 지하철역 개수



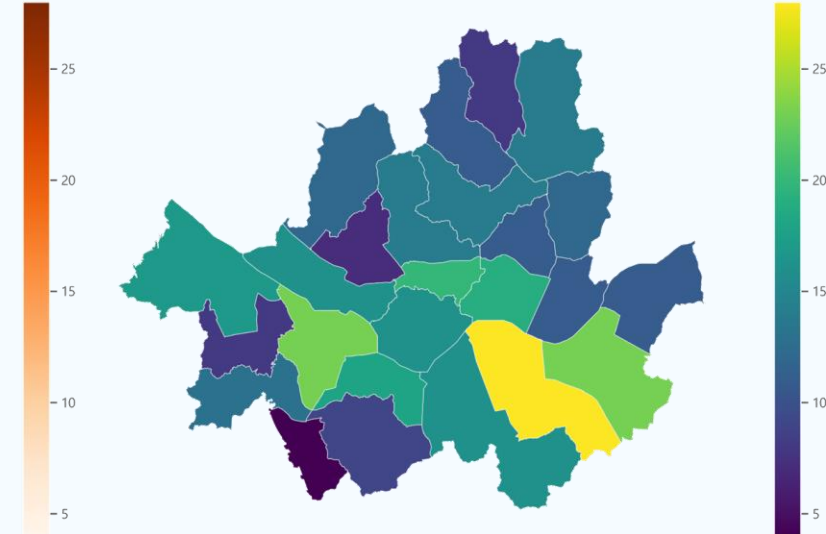
서울시 구별 2021년 지하철역 개수



서울시 구별 2022년 지하철역 개수



서울시 구별 2023년 지하철역 개수



## 데이터 전처리

# 아파트 기준

버스 정류장, 학교, 병원, 공원  
연도별 지하철역

## 거리 계산

## 점수 계산

0~500m 10점 (도보 약 5분)

500m~1km 5점 (도보 약 10분)

1km~1.5km 1점 (도보 약 15분 이상)

## 구/행정동 매핑

동별 직전 1/3/6개월 거래 건수 집계

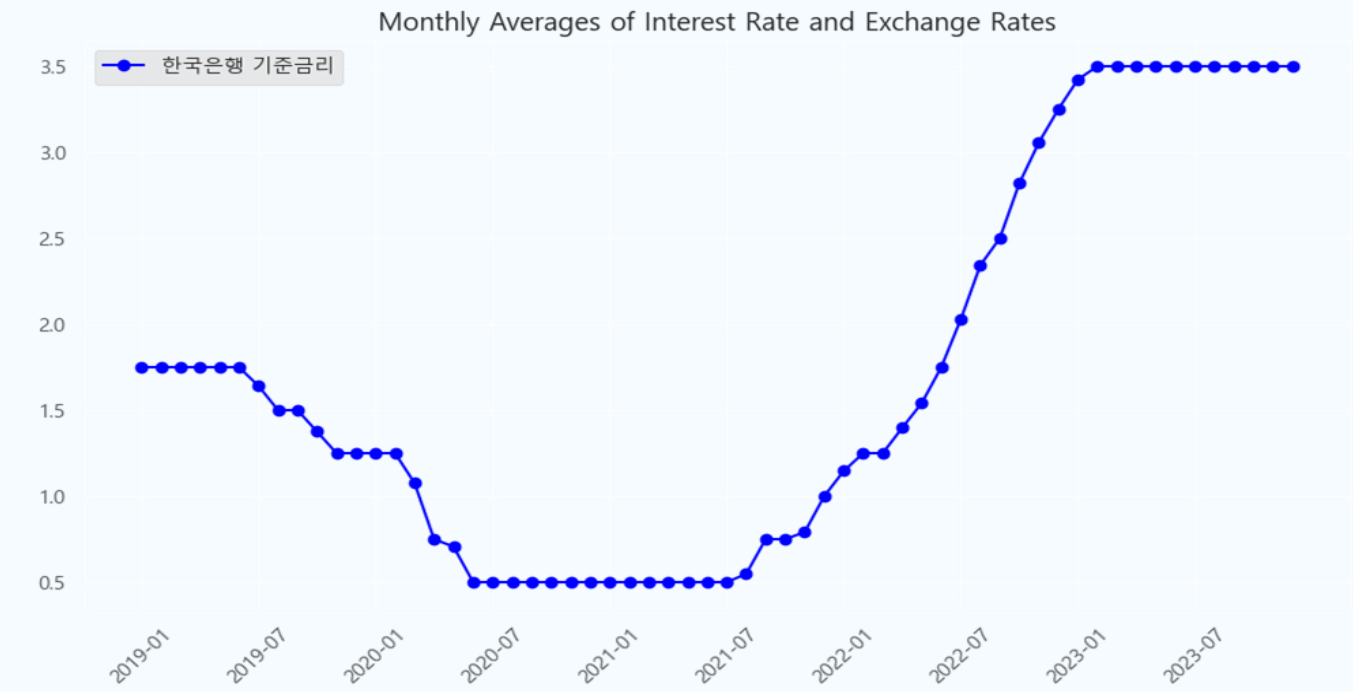
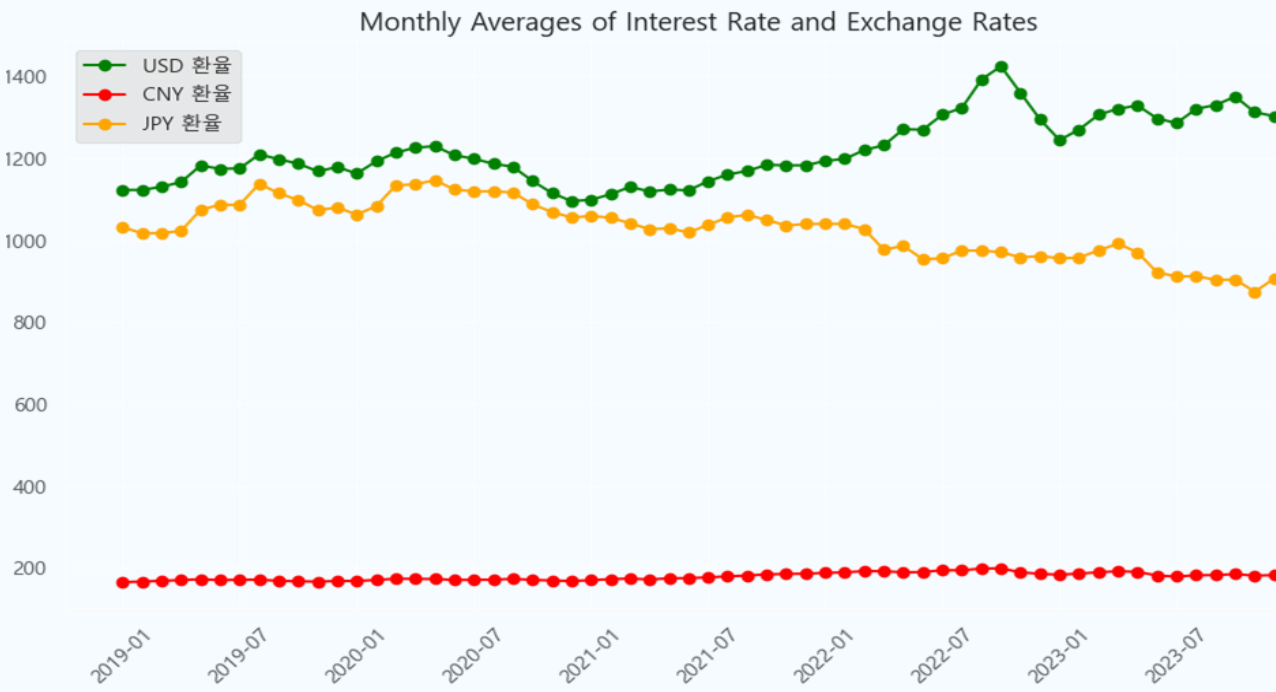
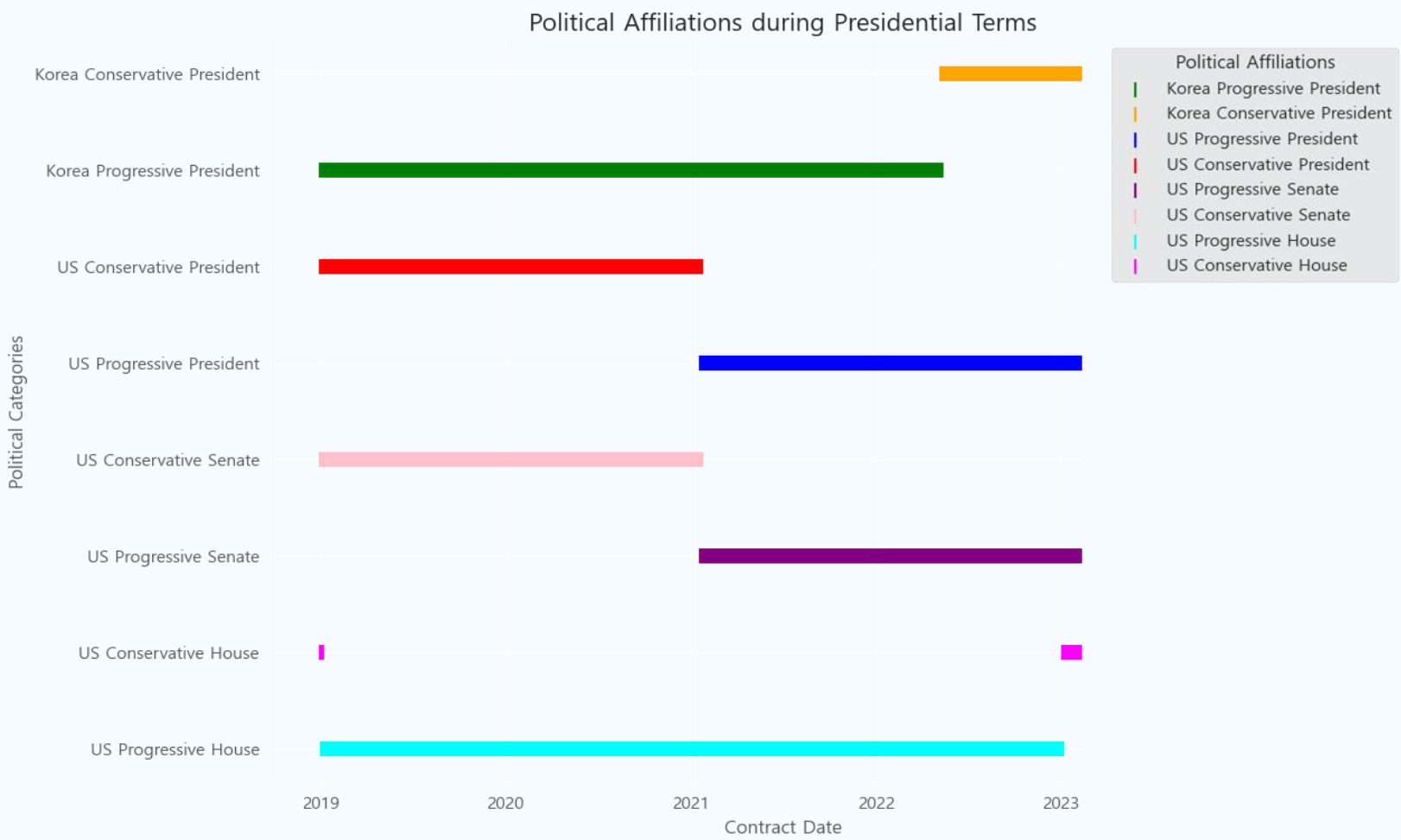
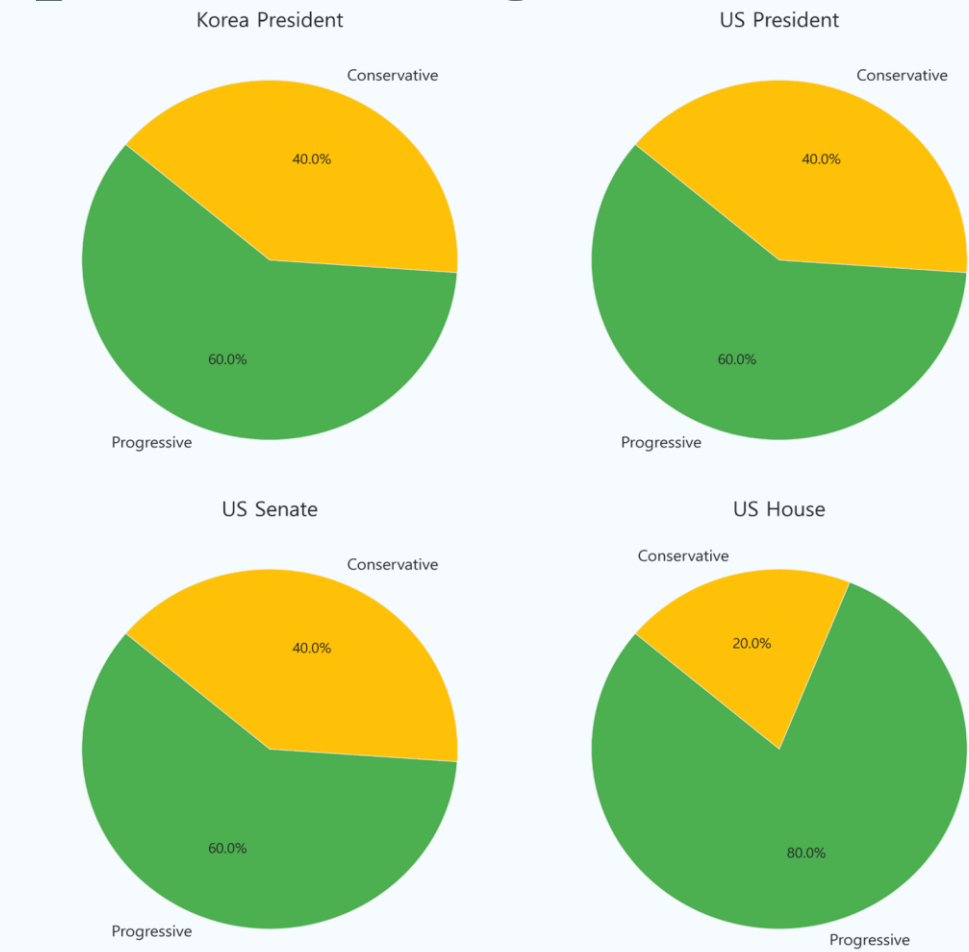
행정동별 인구밀도 데이터 적용

행정동별 상가기준 소득-소비 데이터 적용

# 4. Data Preprocessing

한.미  
정치

환율  
금리



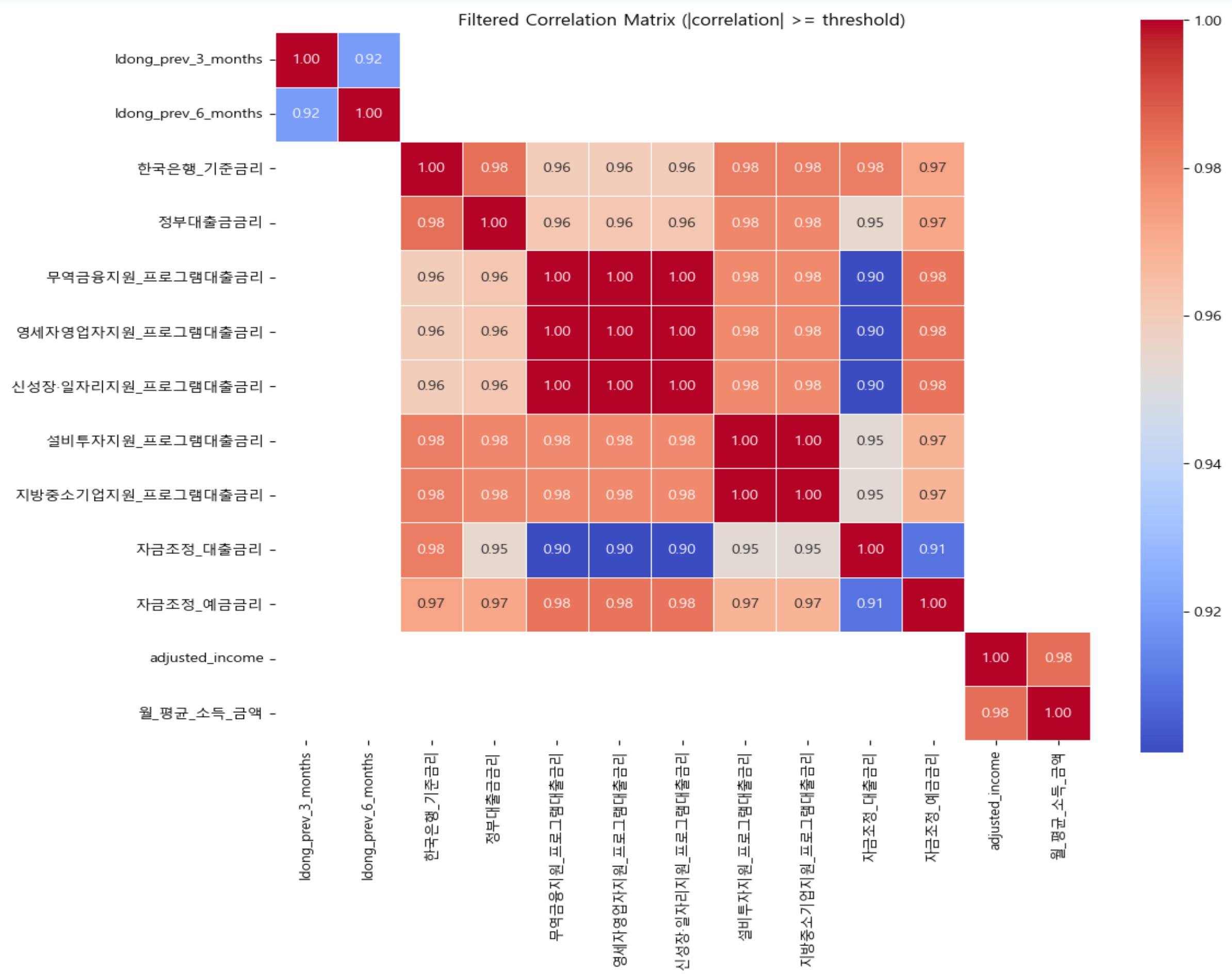
# 5. Project Outcomes

## Modeling

수행결과 - 모델링

# Correlation Analysis

# 상관 관계 분석

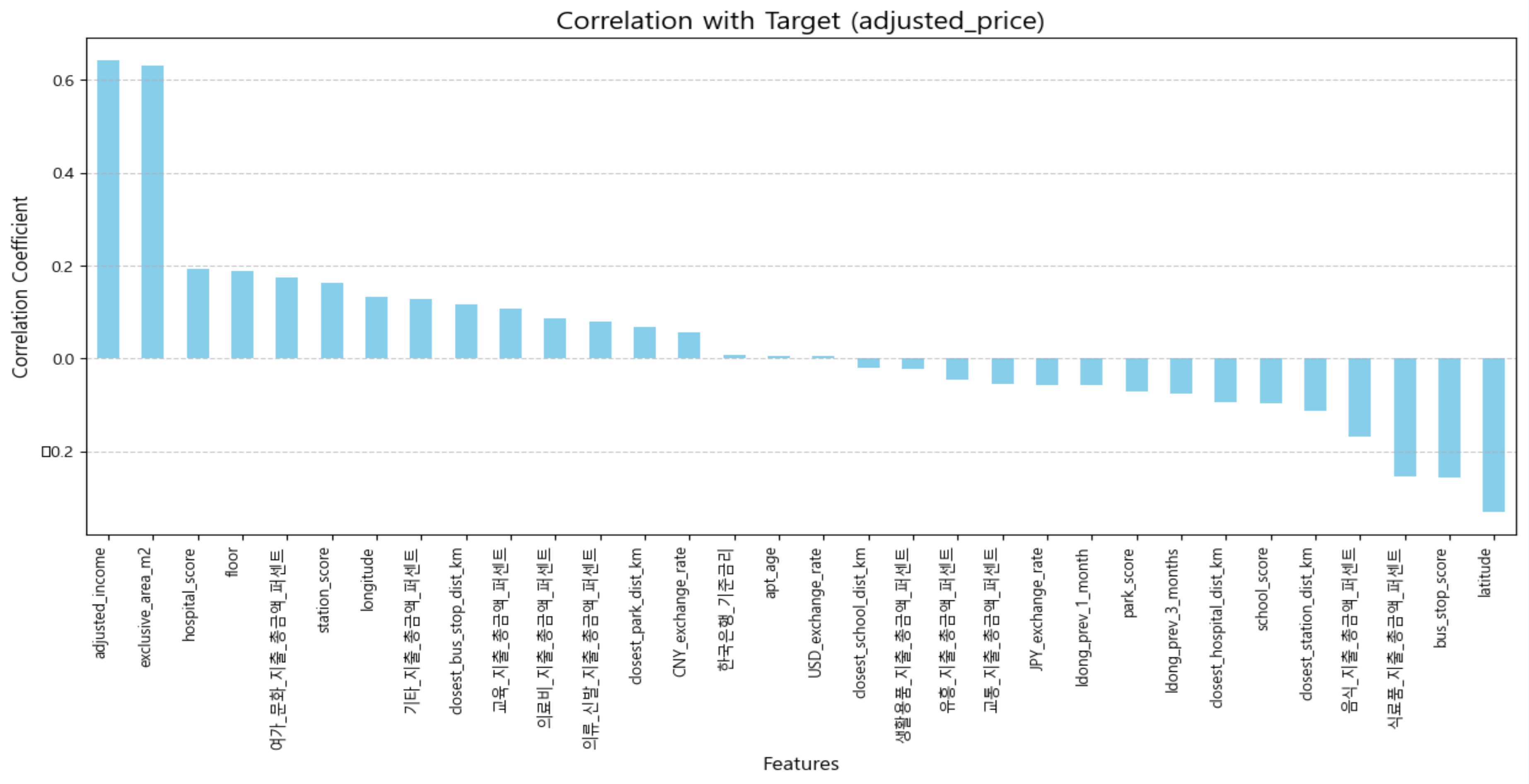


특성 간 상관관계 분석을 통한 다중공선성 탐지

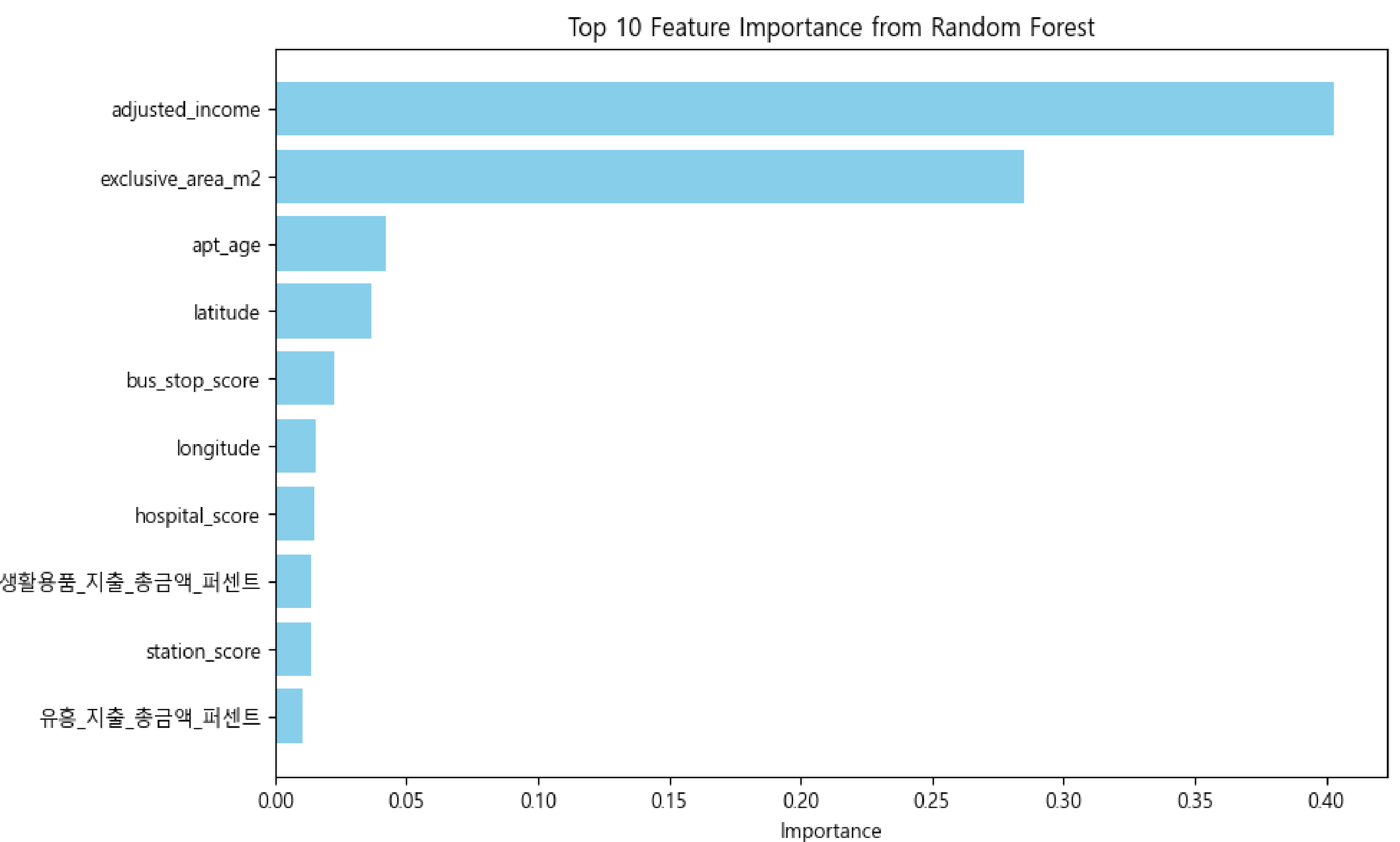


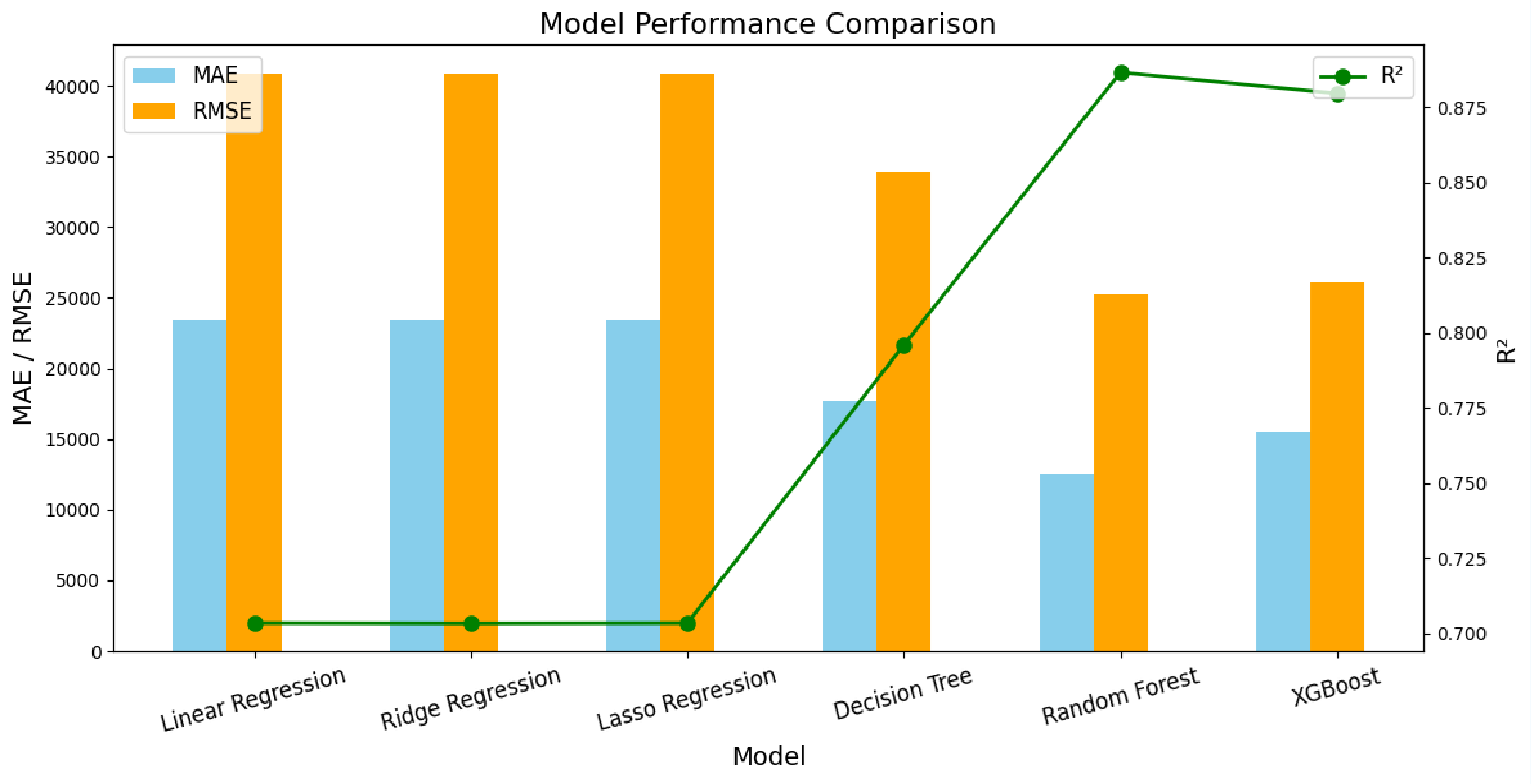
# Correlation Analysis

# 상관 관계 분석



# Feature Importance





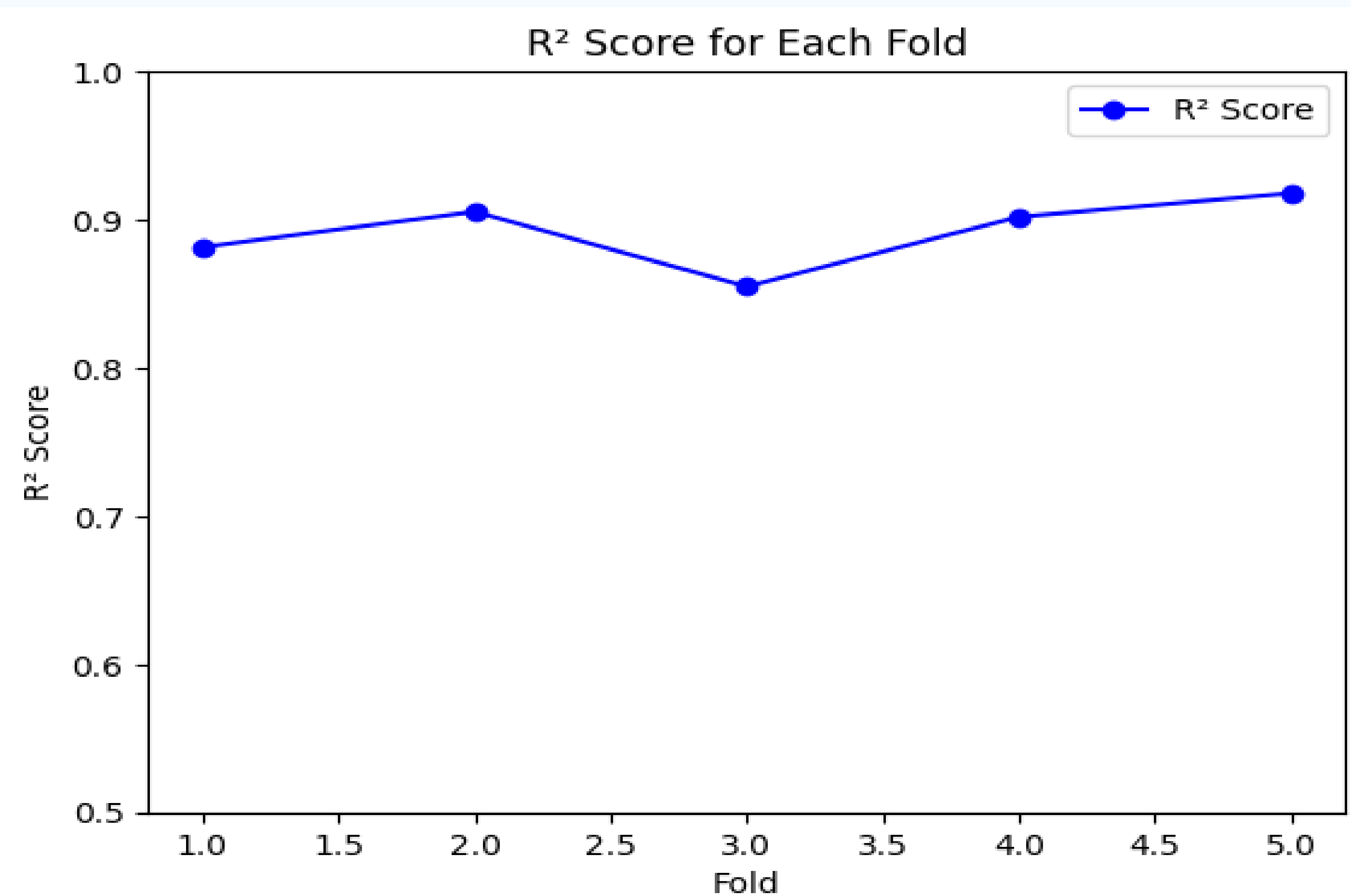


# RandomForestRegressor

랜덤 포레스트	• R <sup>2</sup>	훈련 세트 R <sup>2</sup> : 0.984 테스트 세트 R <sup>2</sup> : 0.871
	• MAE	12731.622
	• 평균 절대 오차	
Error Rate	• MSE	724951722.668
	• 평균 제곱 오차	
	• RMSE	26924.928
0.13	• 평균 제곱근 오차	

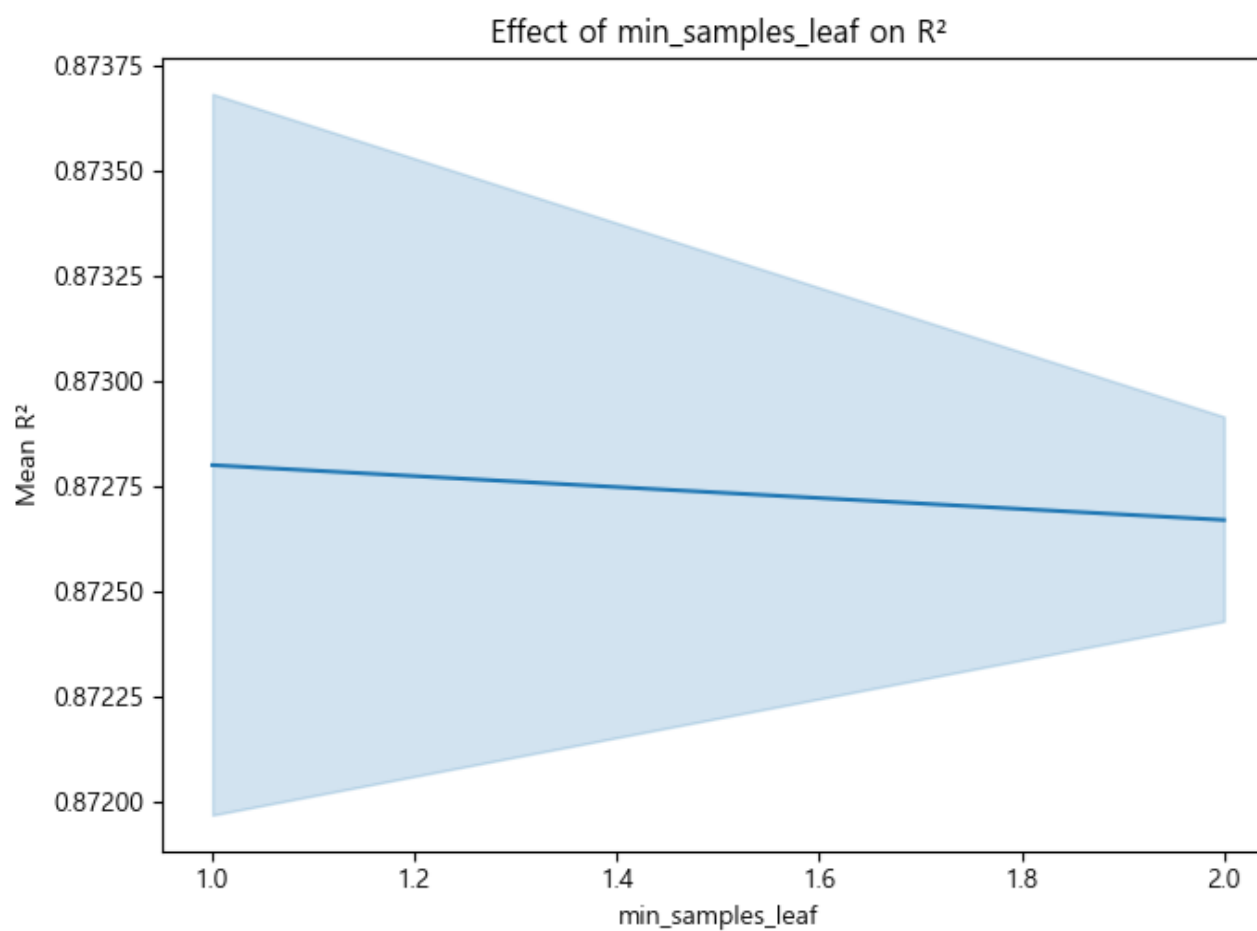
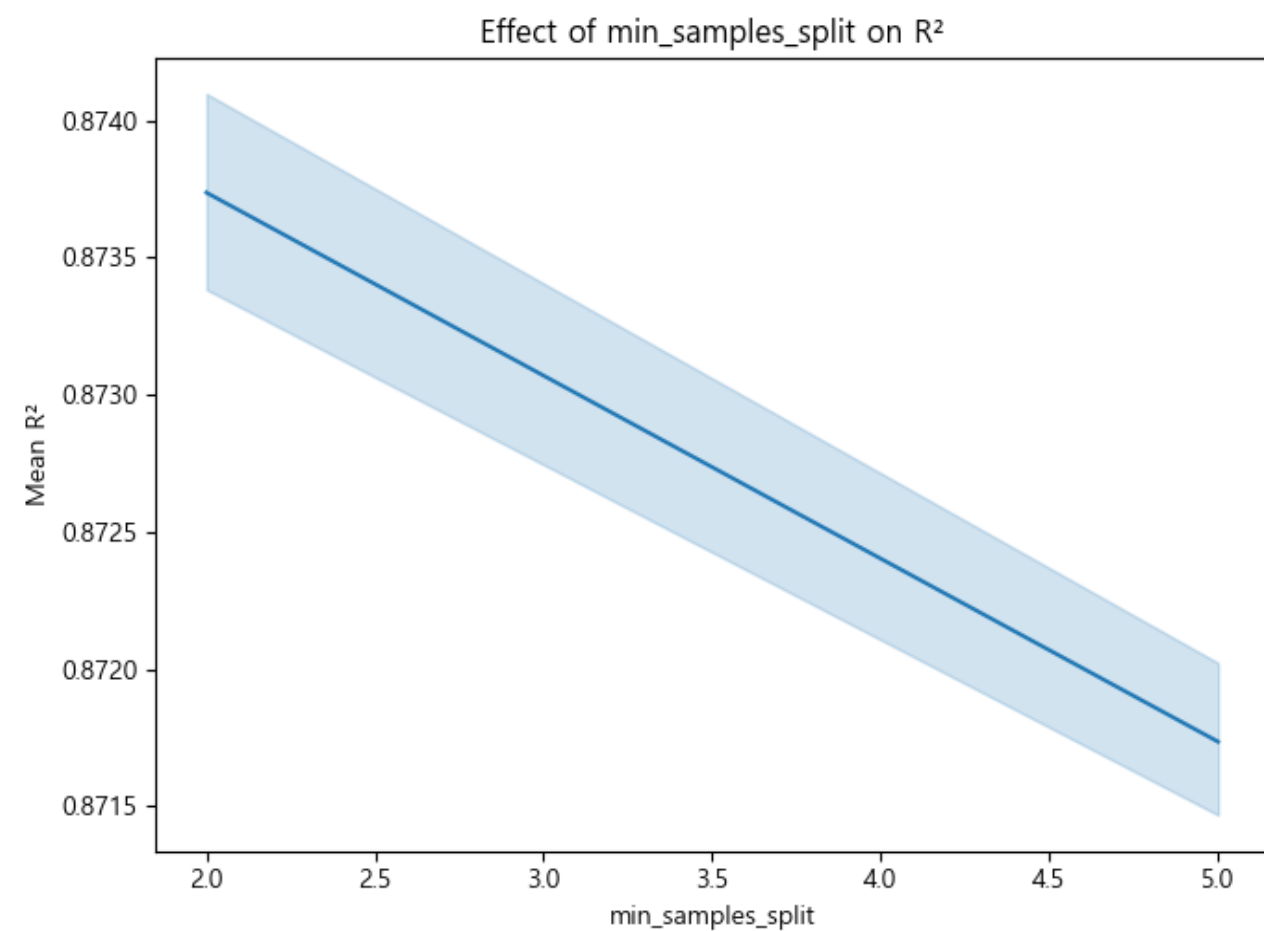
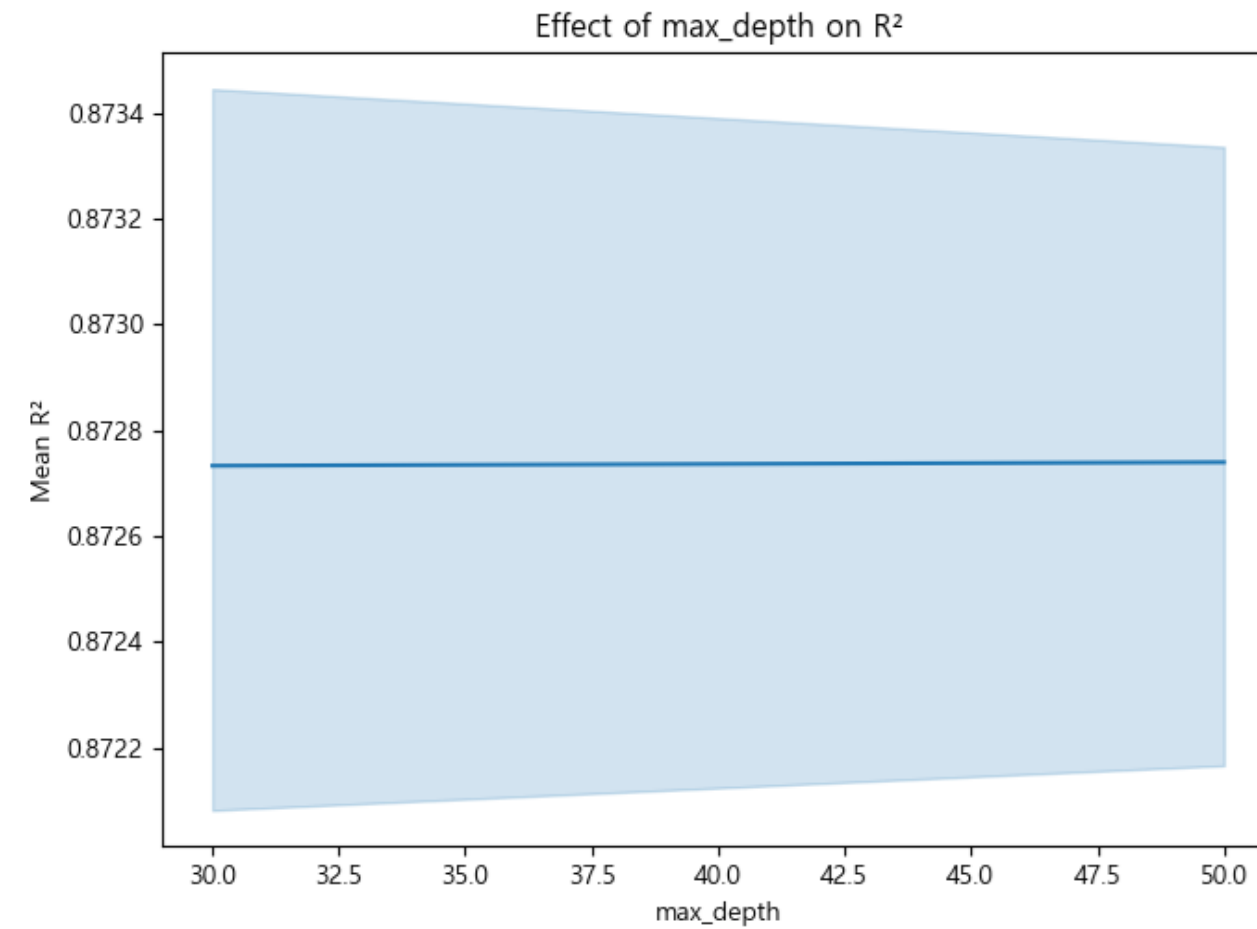
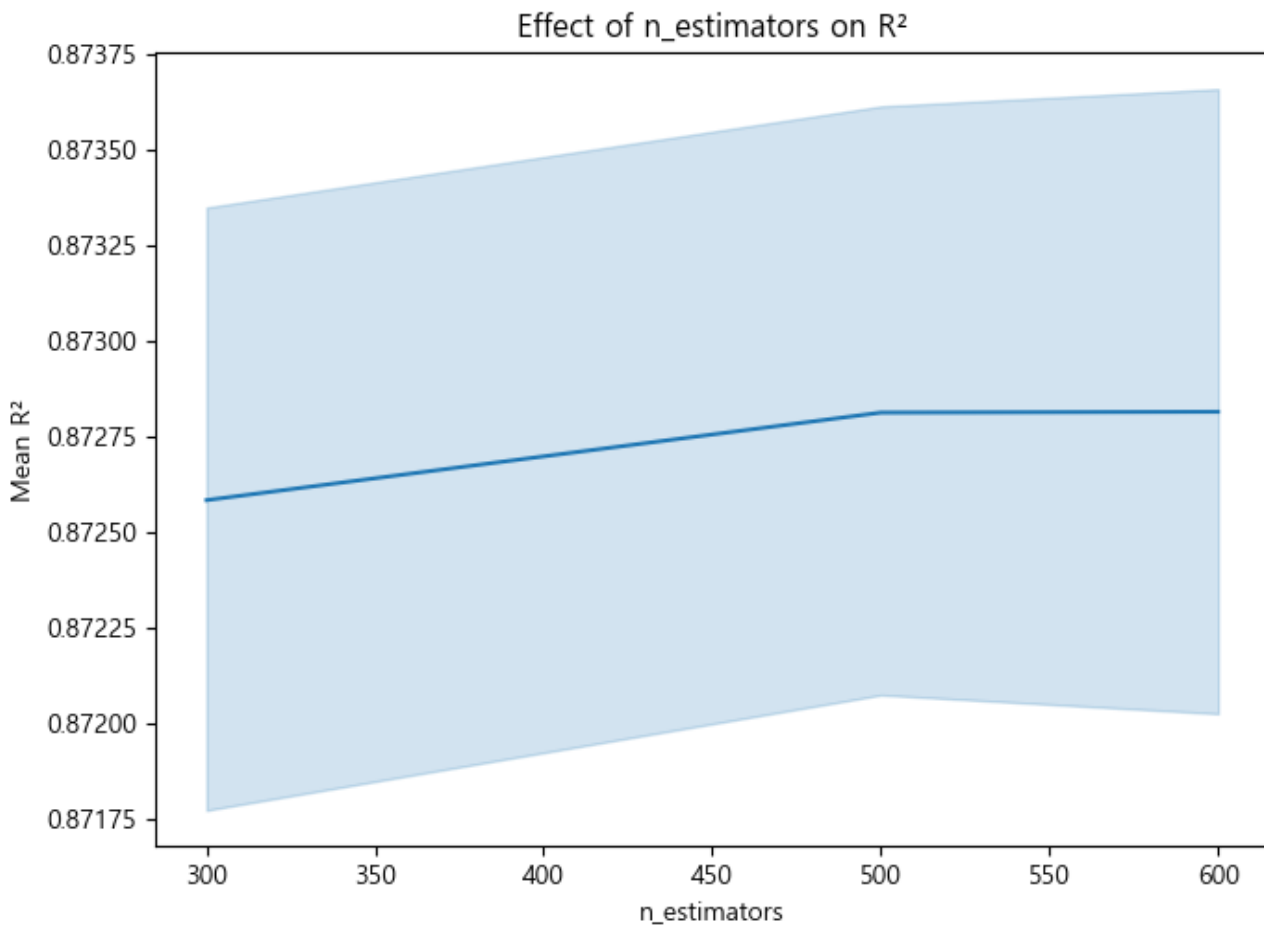
# Cross Validation

교차 검증



# Hyperparameter tuning

## 하이퍼 파라미터 조정



**R<sup>2</sup> = 0.875**

**MAE = 12632.60**

**RMSE = 26503.194**

**Error Rate = 0.129**

# Unsampled Data

•  $R^2$  0.891

• MAE 12170.569  
• 평균 절대 오차

## 랜덤 포레스트

• MSE 540423738.4162  
• 평균 제곱 오차

Error Rate  
0.128

23247.015  
• RMSE  
• 평균 제곱근 오차

# XGBRegressor

• **R<sup>2</sup>**

훈련 세트 R<sup>2</sup> : 0.982  
테스트 세트 R<sup>2</sup> : 0.936

• **MAE**

6735.414

• **평균 절대 오차**

# XGBoost

• **MSE**

89093354.404

• **평균 제곱 오차**

오차율 0.105

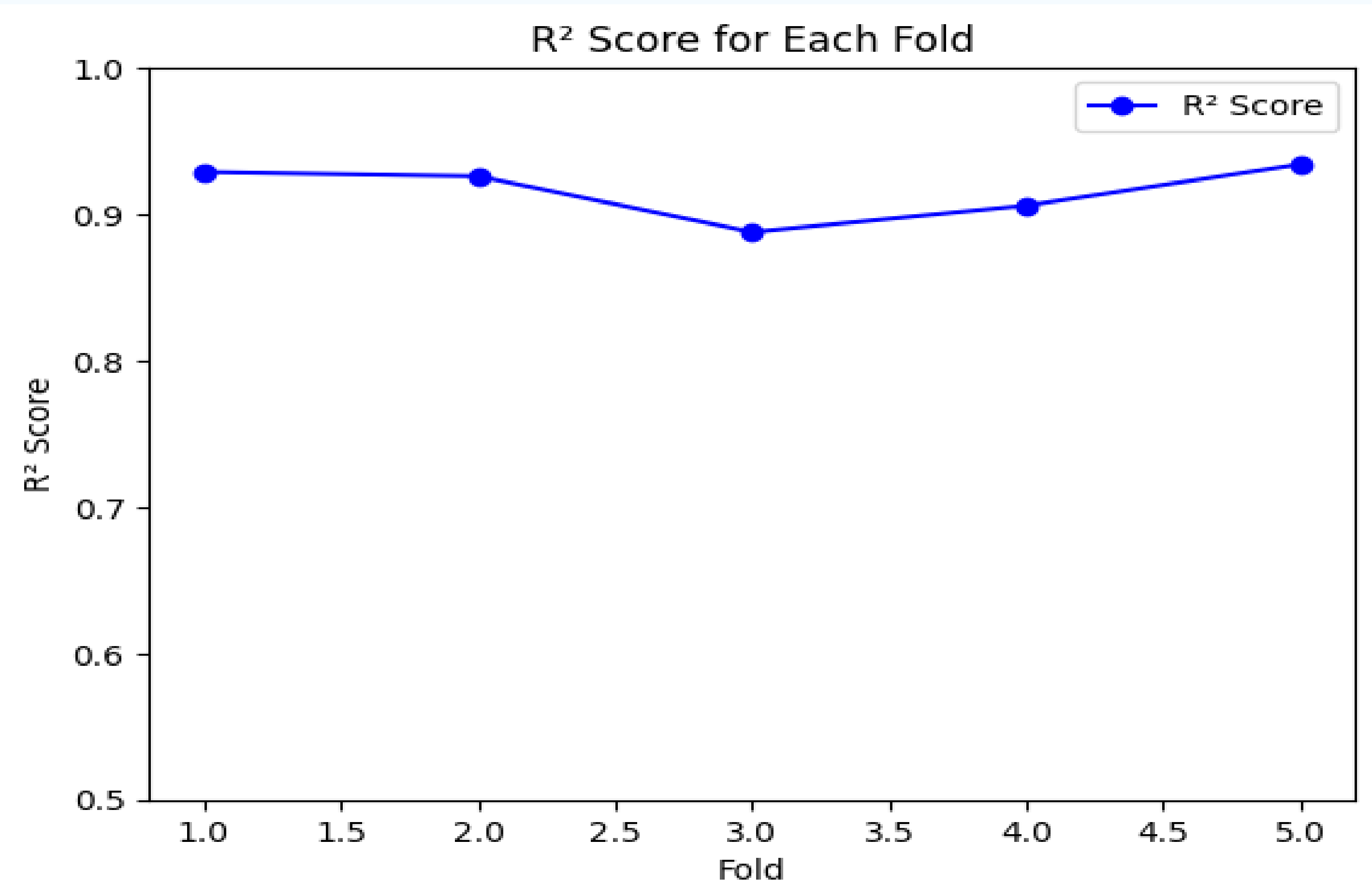
9438.927

• **RMSE**

• **평균 제곱근 오차**

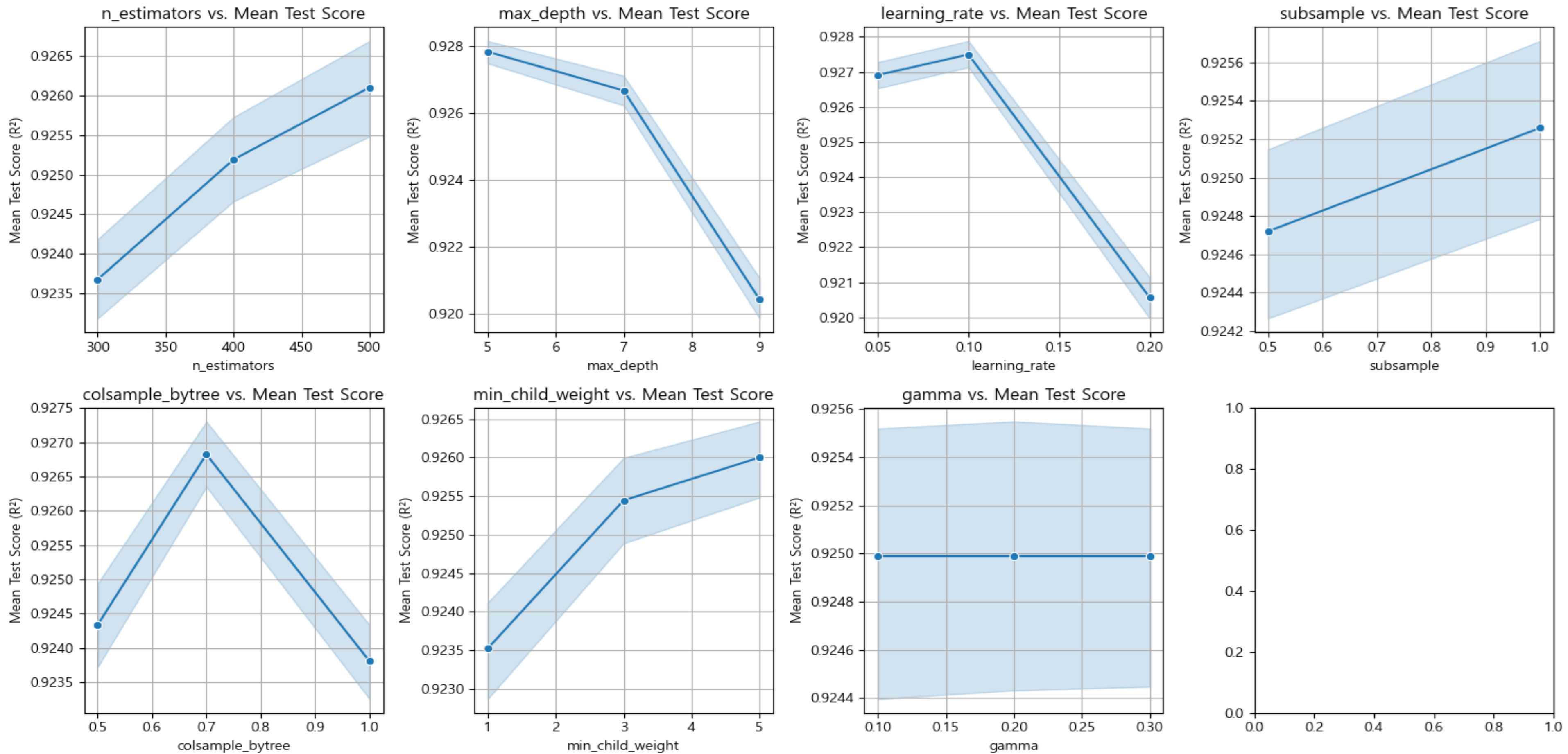
# Cross Validation

교차 검증



# Hyperparameter tuning

## 하이퍼 파라미터 조정



# Unsampled Data

• R<sup>2</sup> 0.936

• MAE 9506.353  
• 평균 절대 오차

## XGBoost

• MSE 316722287.767  
• 평균 제곱 오차

오차율 0.096

17796.693  
• RMSE  
• 평균 제곱근 오차



## 6. Self-Evaluation and Insights

## 자체 평가 의견

### All for one, one for all

각자 담당 데이터를 수집, 전처리  
각자 담당 모델을 훈련  
최고 성능의 모델을 조정



### 이유리

아쉬운 점 :

시간적 여유가 더 있었다면 더 긴 기간 대상 가능

좋았던 점 :

데이터를 전처리하는 다양한 유형을 학습  
다양한 알고리즘의 활용

### 우병준

아쉬운 점 :

시간이 더 있었다면 더 넓은 범위의 파라미터 조정 가능

좋았던 점 :

다양한 알고리즘의 활용  
머신러닝에 대한 보다 깊은 이해

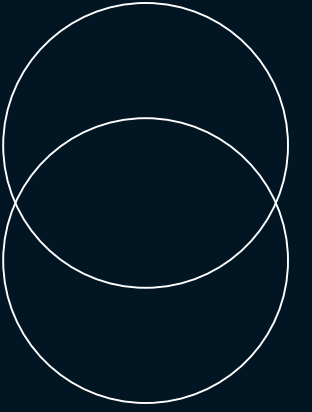
### 이정인

아쉬운 점 :

파이썬이 미숙해 다양한 작업을 하지 못 함

좋았던 점 :

데이터 수집에 대한 이해



Thank You