



# 서울특별시 공공자전거

대여소 수요예측과  
편중지역 관리방안 모색

# 목 차

## 서 론

## 본 론

1. 데이터 전처리(EDA)
2. 모델링 및 결과

## 결 론

# '따릉이' 서울 시민이 가장 좋아하는 정책

## 2017년 서울시민이 뽑은 10대 정책...1위는 '따릉이'

□ 2017년 서울 전역에서 '따릉이' 2만대 시대를 열어갑니다

○ (서비스 확대) "이제 서울 어디에서나 더 가까이 즐기세요!"

- 대여소도 기존 11개 자치구 450개소에서 25개 전 자치구 1,300개소로 늘어납니다. 서울시민 1만 명 당 따릉이 수도 15대에서 20대로 확대됩니다. 특히 대중교통과 갈아타기 용이하도록 지하철역 출입구, 버스·택시승강장의 10~20m 이내, 대여소간 간격은 500m 이내로 설치될 예정입니다.



공공자전거 따릉이', 2년 연속 '내 삶을 바꾼 서울시 10대 뉴스' 1위

비즈니스포스트 | 2018.12.16. | [🔗](#)

5%)를 얻어 지난해에 이어 2년 연속으로 1위를 차지했다고 16일 밝혔다. **서울시는 서울시** 주요 정책 30개를 대상으로 '내 삶을 바꾼 2018 서울시 10대 뉴스' 시민투표를 진행한 결과 '따릉이'가 1만8676표(6.5%)를 얻어...

- ↳ **서울시민이 뽑은 '올해 내 삶을 바꾼 ...** 시민일보 | 2018.12.16.
  - ↳ **서울 시민이 2년 연속으로 뽑은 '공 ...** 한겨레 PICK | 2018.12.16. | 네이버뉴스
  - ↳ **시민 뽑은 서울시 10대뉴스, 2년연 ...** 뉴시스 | 2018.12.16. | 네이버뉴스
  - ↳ **서울시, 시민이 뽑은 10대 뉴스 2년 ...** 매일일보 | 2018.12.16.
- 관련뉴스 17건 전체보기 >

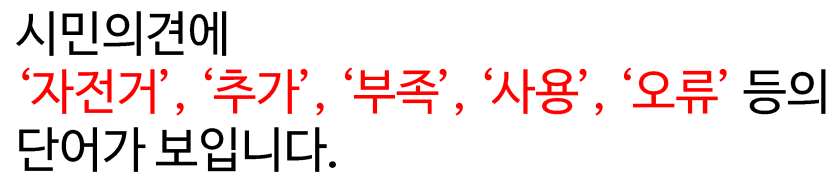


서울시 공유정책 인기는 '따릉이', 활성화는 '주차' 1위

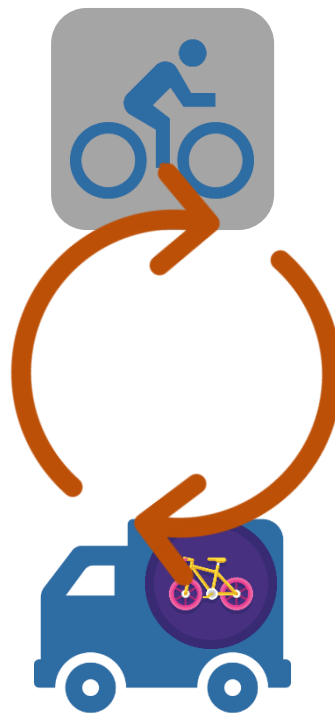
뉴시스 | 2019.01.10. | 네이버뉴스 | [🔗](#)

유희 주차 공간을 공유하는 **서울시 정책**이다. **공유 정책**을 모르는 시민 대상으로 이용 의향이 있는 정책에서도 1위(62.5%)를 차지했다. 이어 ▲공공자전거 **따릉이**(93.1%) ▲공공 공간, 시설 **공유**(90%) ▲공공 대여소(88.7%, ...

- ↳ **서울시민 공유도시 활성화 기대 1위 ...** 머니투데이 | 2019.01.10. | 네이버뉴스
- ↳ **서울시, 성인남녀 대상 '2018년도 공 ...** 환경미디어 | 2019.01.10.



# 따릉이가 부족하다?



## 편도에 의한 편중

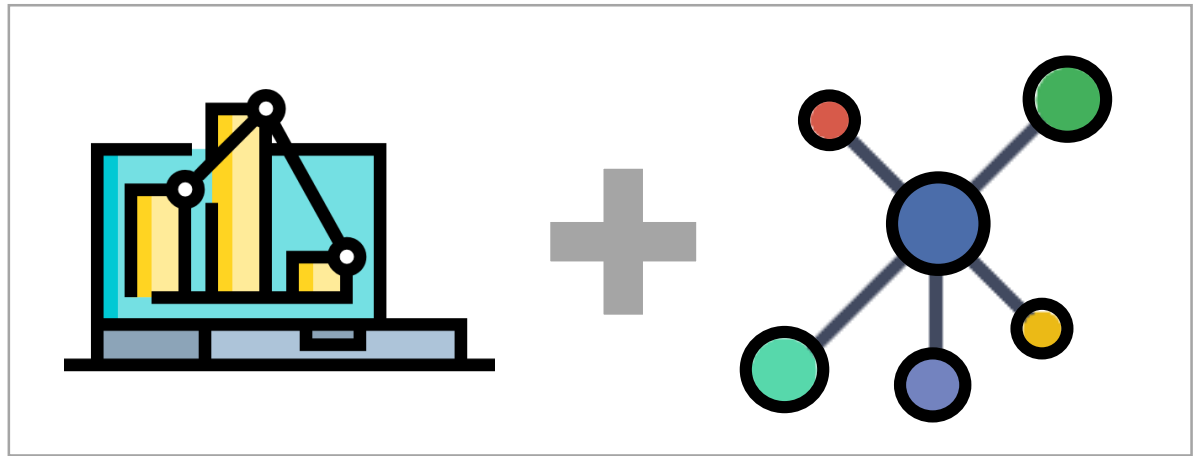
따릉이는 편도 방식이기 때문에 **편중 지역**이 발생합니다.  
이러한 편중 지역 때문에  
대여가 많은 곳은 따릉이가 매번 부족하고, 반납이 많은 곳은 따릉이가 쌓이게 됩니다.

## 따릉이 분배업무

서울시설공단 총 85~143명의 직원들, 분배 트럭 총 66대로 편중 문제를 해소하고 있습니다.

이는 시민에게 이용시 **불편**을 주고,  
관리자에게는 **관리 비용**을 발생시킨다.

빅데이터로  
해결  
하자



위치, 요일, 시간대 등을 고려하여  
수요예측분석과 네트워크 분석으로  
따릉이 편종을 해소해보자!

**01**

# **데이터 전처리 (EDA)**

# 01 EDA 원데이터의 구성

- 1) 원데이터는 ‘서울 열린데이터 광장’에서 제공하는 ‘서울특별시 공공자전거 대여이력정보’, ‘서울특별시 공공자전거 대여소 정보’ 를 2017년 10월부터 2018년 9월까지의 데이터를 사용하였습니다.
- 2) ‘서울특별시 공공자전거 대여이력 정보’ 데이터에서는 대여일시, 대여소 번호, 반납일시, 반납대여소 번호를 사용하였습니다.
- 3) ‘서울특별시 공공자전거 대여소 정보’ 데이터에서는 대여소 번호, 위도, 경도를 사용하였습니다.



서울특별시 공공자전거  
대여이력 정보

자전거번호	대여일시	대여소번호	대여소명	거치대번호	반납일시	반납대여소번호	반납대여소명	반납 거치대번호	이용시간(분)	이용거리(M)
SPB-00230	2017-01-01 0:00	419	홍플러스 앞	5	2017-01-01 0:21	914	새절역 2번출구	18	20	3340
SPB-02803	2017-01-01 0:01	825	서빙고동 주민센터	2	2017-01-01 0:15	822	이촌1동 마을공	7	14	3380
SPB-03956	2017-01-01 0:05	906	연신내역 5번출	1	2017-01-01 0:16	931	역촌파출소	6	11	1610
SPB-02864	2017-01-01 0:05	906	연신내역 5번출	9	2017-01-01 0:29	912	응암오거리	1	24	3620
SPB-01388	2017-01-01 0:05	330	청계천 한빛광장	6	2017-01-01 0:23	171	임광빌딩 앞	9	17	2280
SPB-04437	2017-01-01 0:06	347	동대문역사문화	16	2017-01-01 0:18	344	성균관대 E하우	4	11	1380
SPB-04461	2017-01-01 0:06	334	종로3가역 2번출	9	2017-01-01 0:29	354	포르투갈 대사관	10	22	2430
SPB-03383	2017-01-01 0:06	175	홍연2교 옆	1	2017-01-01 1:07	175	홍연2교 옆	1	60	12150



서울특별시 공공자전거  
대여소 정보

구명	대여소ID	대여소번호	대여소명	대여소 주소	거치대수	위도	경도
마포구	ST-3	101	(구)합정동 주민센터	시 마포구 동교로8길 5	5	37,549561	126,905754
마포구	ST-4	102	망원역 1번출구 앞	특별시 마포구 월드컵로 74	20	37,555649	126,910629
마포구	ST-5	103	망원역 2번출구 앞	특별시 마포구 월드컵로 74	14	37,554951	126,910835
마포구	ST-6	104	합정역 1번출구 앞	특별시 마포구 양화로 59	13	37,550629	126,914986
마포구	ST-7	105	합정역 5번출구 앞	특별시 마포구 양화로 48	5	37,550007	126,914825
마포구	ST-8	106	합정역 7번출구 앞	특별시 마포구 독막로 44	10	37,548645	126,912827
마포구	ST-9	107	한은은행 서교동금융센터점	특별시 마포구 월드컵북로 3	5	37,55751	126,918503



## - DATA CLEANSING

- ✓ 서비스를 중단 대여소, 정비소 등 분석에 무관한 데이터 삭제
- ✓ 값이 없는 위도 경도 데이터 직접 입력

## - 변수 생성(평일/휴일)

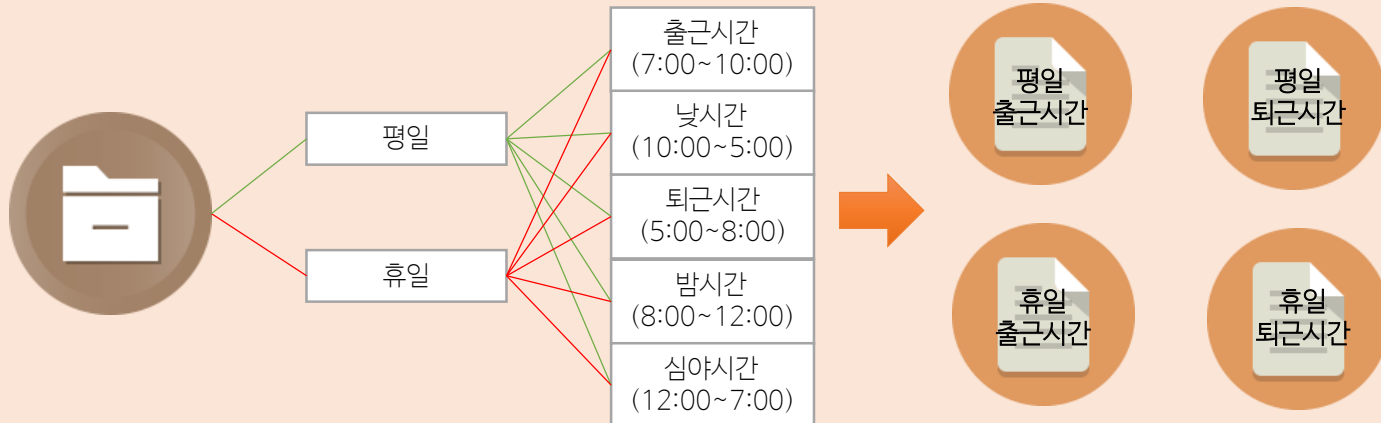
변수명	변수 설명	value
weekday	요일	0~6
weekend	주말	TRUE/FARSE
holiday	공휴일	TRUE/FARSE

## - 변수 생성(대여수, 반납수)

변수명	변수 설명	value
대여수	시간대별로 한 대여소에 대여한 횟수를 COUNT	int
반납수	시간대별로 한 대여소에 반납된 횟수를 COUNT	int

## - 데이터 셋 분리(평일/주말/휴일, 시간대로) → 프로젝트의 목적에 따라 평일의 **출근시간**과 **퇴근시간**만을 분석

- ✓ 데이터 셋을 평일(red)과 휴일(nonred)로 나눈다.
- ✓ 이를 시간대별로 출근시간, 낮시간, 퇴근시간, 밤시간, 심야시간으로 나눈다.



# 01\_EDA 최종 분석 데이터



소셜네트워크 분석 데이터셋

101	102	103	104	105	106	107	108	109	110
101	29	26	16	219	58	70	105	18	10
102	33	39	5	21	5	22	908	93	6
103	5	10	25	25	22	16	110	84	9
104	20	19	7	21	3	6	65	38	3
105	3	2	5	2	16	5	2	185	182
106	51	41	21	4	5	42	23	16	124
107	0	83	83	63	2	13	51	73	39
108	1	14	1	5	12	5	13	16	2
109	1	44	29	40	31	66	19	14	31
110	0	0	1	1	0	2	8	4	0

최종 분석 데이터

- ✓ 행을 기준으로 대여를,
- ✓ 열을 기준으로 반납을 나타내는

대여소간 이용패턴 행렬 생성

수요예측 데이터셋

	depholiday	depweekend	scaled_PM10	scaled_PM25	scaled_rain	scaled_temp	mon	tue	wed	thu	fri	sat	sun	113_dep
2017-10-01	0.0	1.0	0.263456	0.269531	0.0	0.721024	0.0	0.0	0.0	0.0	0.0	0.0	1.0	4.0
2017-10-02	1.0	0.0	0.076487	0.085938	0.0	0.727763	1.0	0.0	0.0	0.0	0.0	0.0	0.0	7.0
2017-10-03	1.0	0.0	0.195467	0.074219	0.0	0.648922	0.0	1.0	0.0	0.0	0.0	0.0	0.0	7.0
2017-10-04	1.0	0.0	0.113314	0.074219	0.0	0.609838	0.0	0.0	1.0	0.0	0.0	0.0	0.0	2.0
2017-10-05	1.0	0.0	0.059490	0.050781	0.0	0.669137	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2.0
2017-10-06	1.0	0.0	0.084986	0.074219	0.0	0.718329	0.0	0.0	0.0	0.0	1.0	0.0	0.0	7.0
2017-10-07	0.0	1.0	0.019830	0.023438	0.0	0.714960	0.0	0.0	0.0	0.0	0.0	1.0	0.0	3.0
2017-10-08	0.0	1.0	0.076487	0.078125	0.0	0.775606	0.0	0.0	0.0	0.0	0.0	0.0	1.0	6.0
2017-10-09	1.0	0.0	0.141643	0.148438	0.0	0.776280	1.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0
2017-10-10	0.0	0.0	0.243626	0.218750	0.0	0.752022	0.0	1.0	0.0	0.0	0.0	0.0	0.0	27.0

최종 분석 데이터에

- ✓ 기후 변수(미세먼지, 우천, 기온) 추가
  - ✓ 더미 변수(요일별, holiday, weeked) 추가
- 한 데이터 셋 생성

02

**모델링**  
**(결과시각화)**

# 네트워크분석

community detection 알고리즘 중 하나인 walktrap 알고리즘을 사용하여 네트워크 분석을 실행하였다. walktrap 알고리즘은 random walk를 사용하는 대규모 네트워크에 대하여 커뮤니티를 계산하는 방법이며, random walk는 동일한 커뮤니티 내에서 빈번히 이루어질 것이라는 아이디어에서 출발하였다.

random walk란 그래프 상에서 무작위로 어떤 시작 노드부터 이웃 노드로 움직이는 것을 말하는데, random walk를 통해 반복적으로 무한히 이동하다 보면 각 노드를 방문할 확률이 나오게 된다. 이 random walk를 기반으로 양방향 유사성을 계산한다. 이 값이 각 커뮤니티의 정점이 되며, 이를 기반으로 커뮤니티를 생성하게 된다.

R을 이용한 walktrap 알고리즘은 'igraph' 패키지 내의 'walktrap.community' 라이브러리를 사용하여 구현할 수 있다.

## R에서 walktrap.community 사용법

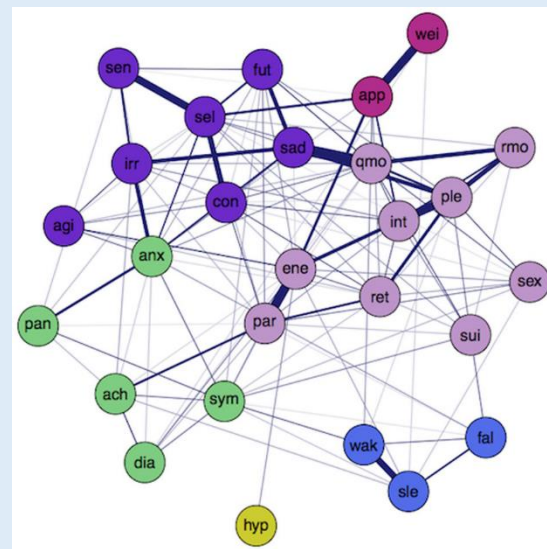
```
walktrap.community(graph, weights = E(graph)$weight, steps = 4, merges =
  TRUE, modularity = FALSE, labels = TRUE)
```

### walktrap.community의 인수 설명

본 분석에서는 'weight'를 각 대여소의 일별/시간대별 총 자전거 대수마다 가중치를 주었으며, 'merges'와 'modularity'는 True로 주었다.

인수명	설명
graph	input graph
weights	각 연결선의 가중치
steps	수행할 random walk의 길이
merges	결과에 병합 행렬을 포함할지 여부를 나타내는 논리적 스칼라
modularity	결과에 모듈성 값의 벡터를 포함할지 여부를 나타내는 논리적 스칼라, membership 인수가 True이면 항상 계산됨
labels	True이고 그래프에 정점의 속성이 있다면, 'labels'라는 목록에 결과로 포함됨

## walktrap 구현 예시



## 수요예측분석

RNN은 히든 노드가 방향을 가진 엣지로 연결돼 순환구조를 이루는(directed cycle) 인공신경망입니다.

LSTM은 RNN의 Hidden Layer를 Input Gate, Output Gate, Forget Gate라는 세가지 게이트로 구성된 Memory Block으로 대체한 구조이며, 긴 의존 기간을 필요로 하는 학습을 수행할 능력을 가진 알고리즘이다.

RNN과 LSTM 모두 시계열 데이터에 효과적이지만, 대부분의 문제에 대해서 LSTM이 RNN보다 좋은 결과를 내는 경향이 있다.

Python에서 RNN과 LSTM은 Keras 라이브러리를 임포트하여 사용 가능하다.

본 분석에는 1차로 RNN, 2차로 polynomialFeatures로 RNN을, 3차로 LSTM을 사용하였습니다.

```
#1 1차
model = Sequential()
model.add(Dense(64, input_dim=13, activation='relu'))
for i in range(3):
    model.add(Dense(64, activation='relu'))
    #model.add(Dropout(0.3))
model.add(Dense(1))

model.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
early_stopping = EarlyStopping(patience= 20)
hist = model.fit(scaled_x, train_y, epochs=3000, batch_size = 10)
```

```
# define model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(n_steps, 13)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
```

```
poly = PolynomialFeatures(degree=2)
scaled_x_poly = poly.fit_transform(scaled_x)
```

```
#2 2차 poly
model = Sequential()
model.add(Dense(64, input_dim=15, activation='relu'))
for i in range(3):
    model.add(Dense(64, activation='relu'))
    #model.add(Dropout(0.3))
model.add(Dense(1))

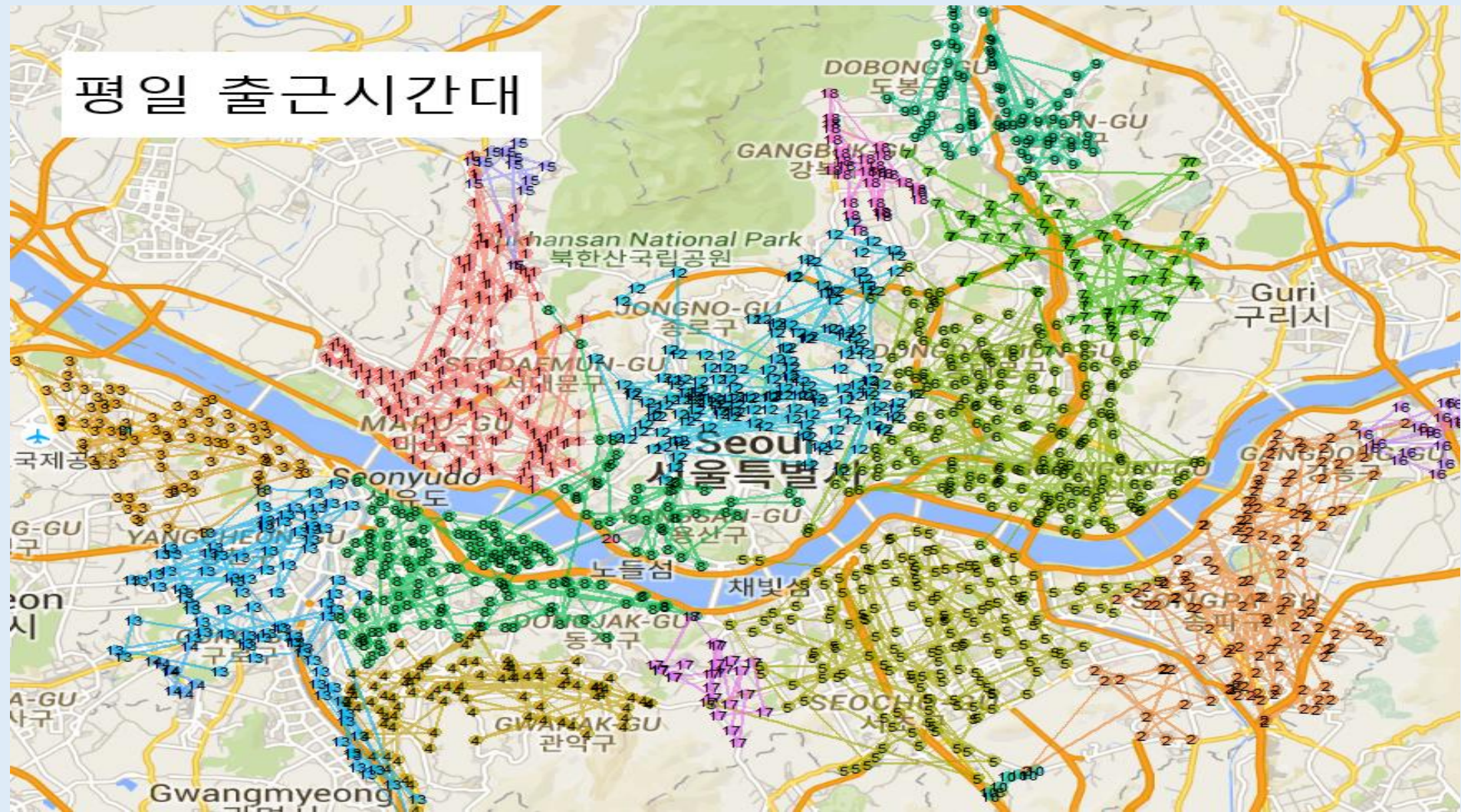
model.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
early_stopping = EarlyStopping(patience= 20)
hist = model.fit(scaled_x_poly, pd.DataFrame(mg_113_1_y), epochs=3000, batch_size = 10)
```



## 02modeling 결과 시각화 \_네트워크 분석

### 평일\_네트워크 분석 결과

평일 출근시간대

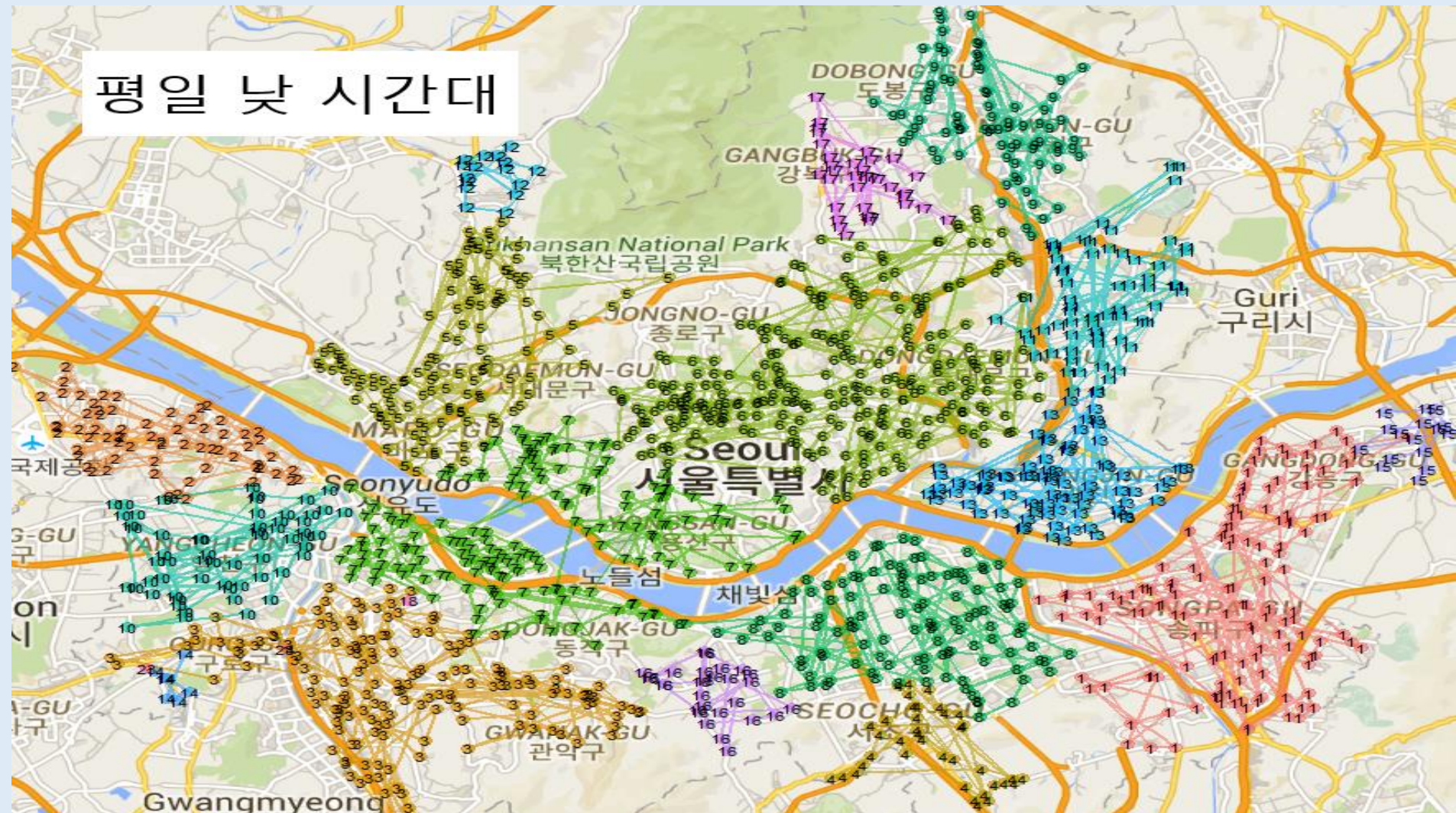




## 02modeling 결과 시각화 \_네트워크 분석

### 평일\_네트워크 분석 결과

평일 낮 시간대

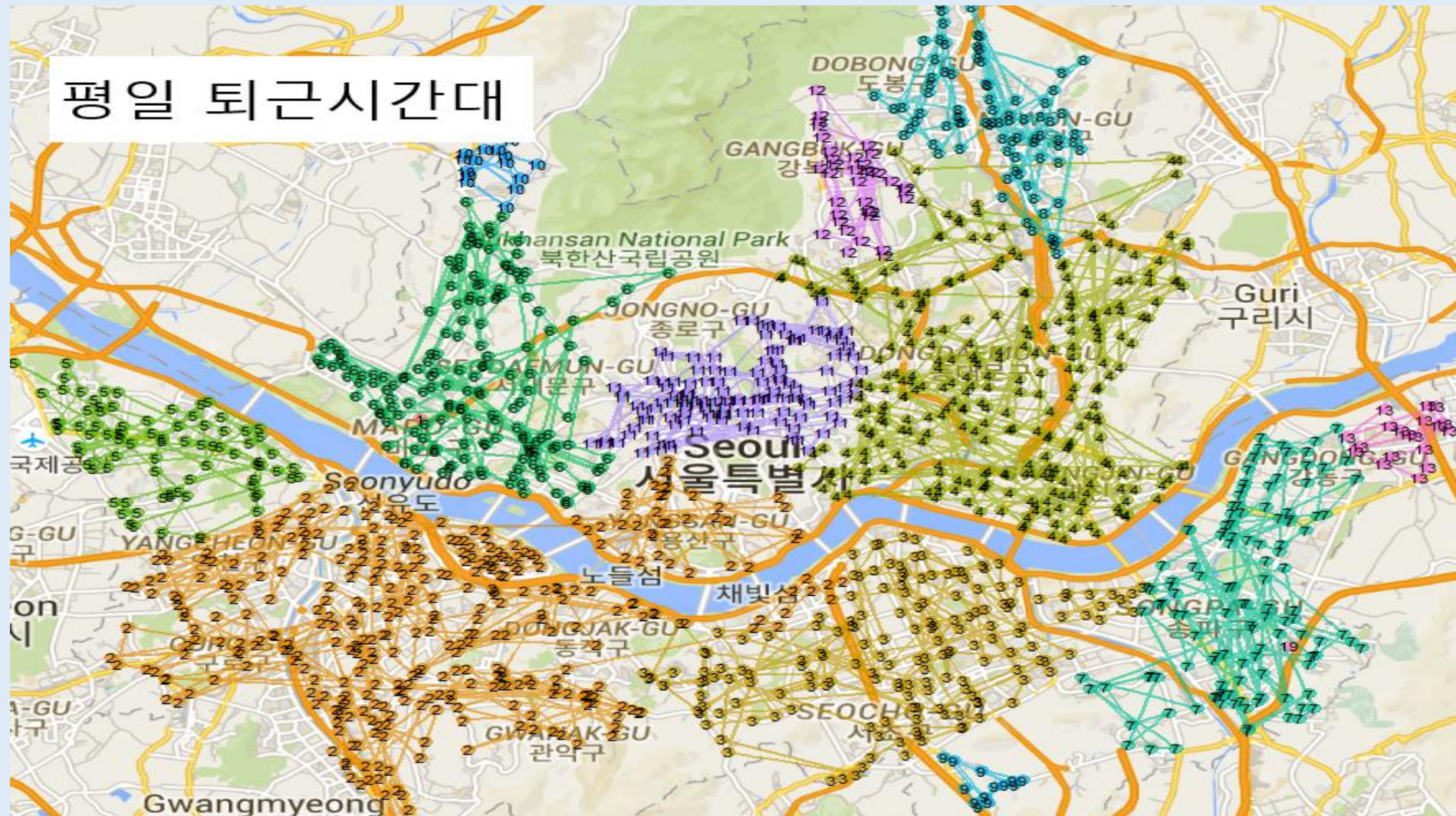




## 02modeling 결과 시각화 \_네트워크 분석

### 평일\_네트워크 분석 결과

평일 퇴근시간대

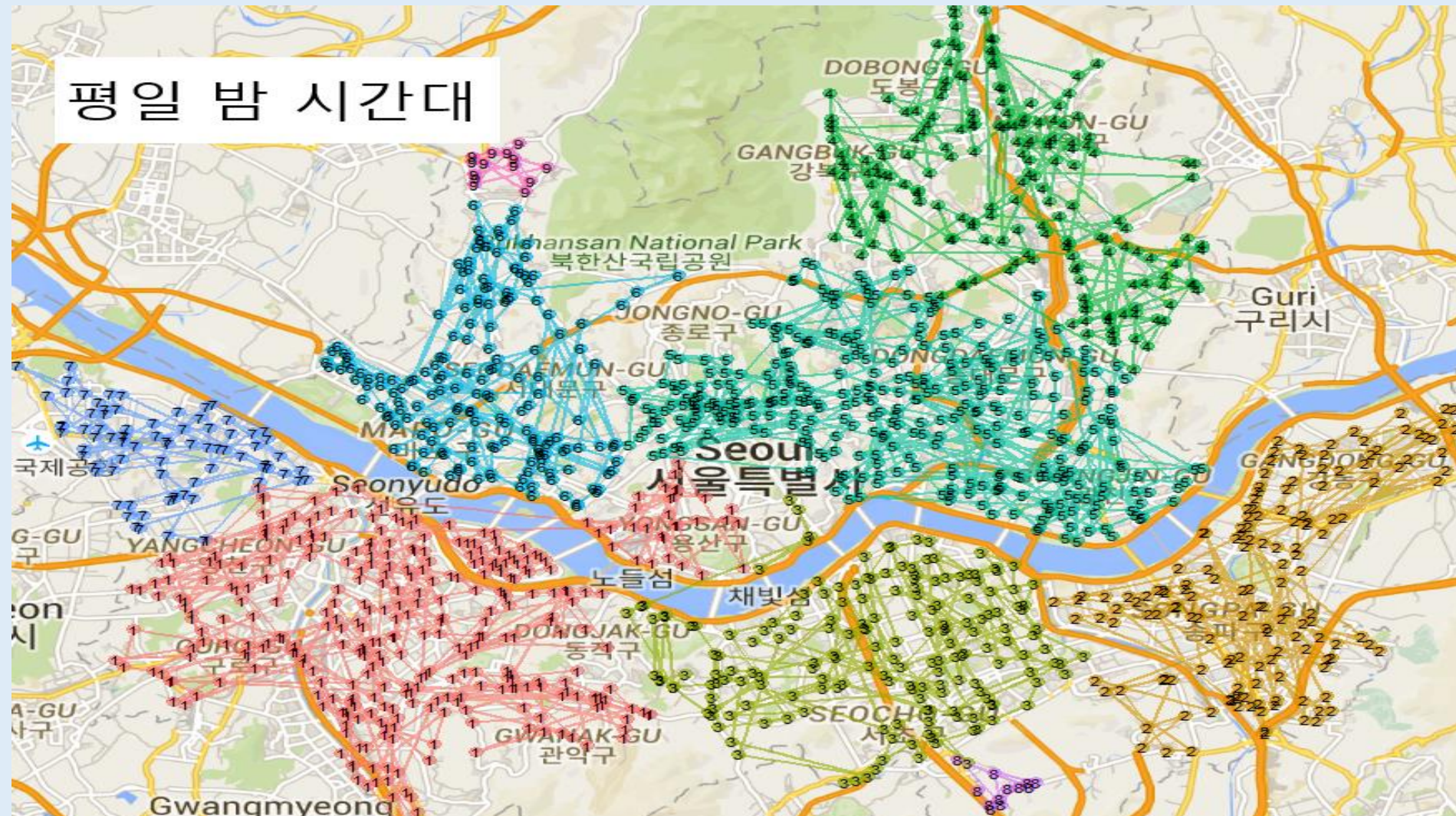




## 02modeling 결과 시각화 \_네트워크 분석

### 평일\_네트워크 분석 결과

평일 밤 시간대

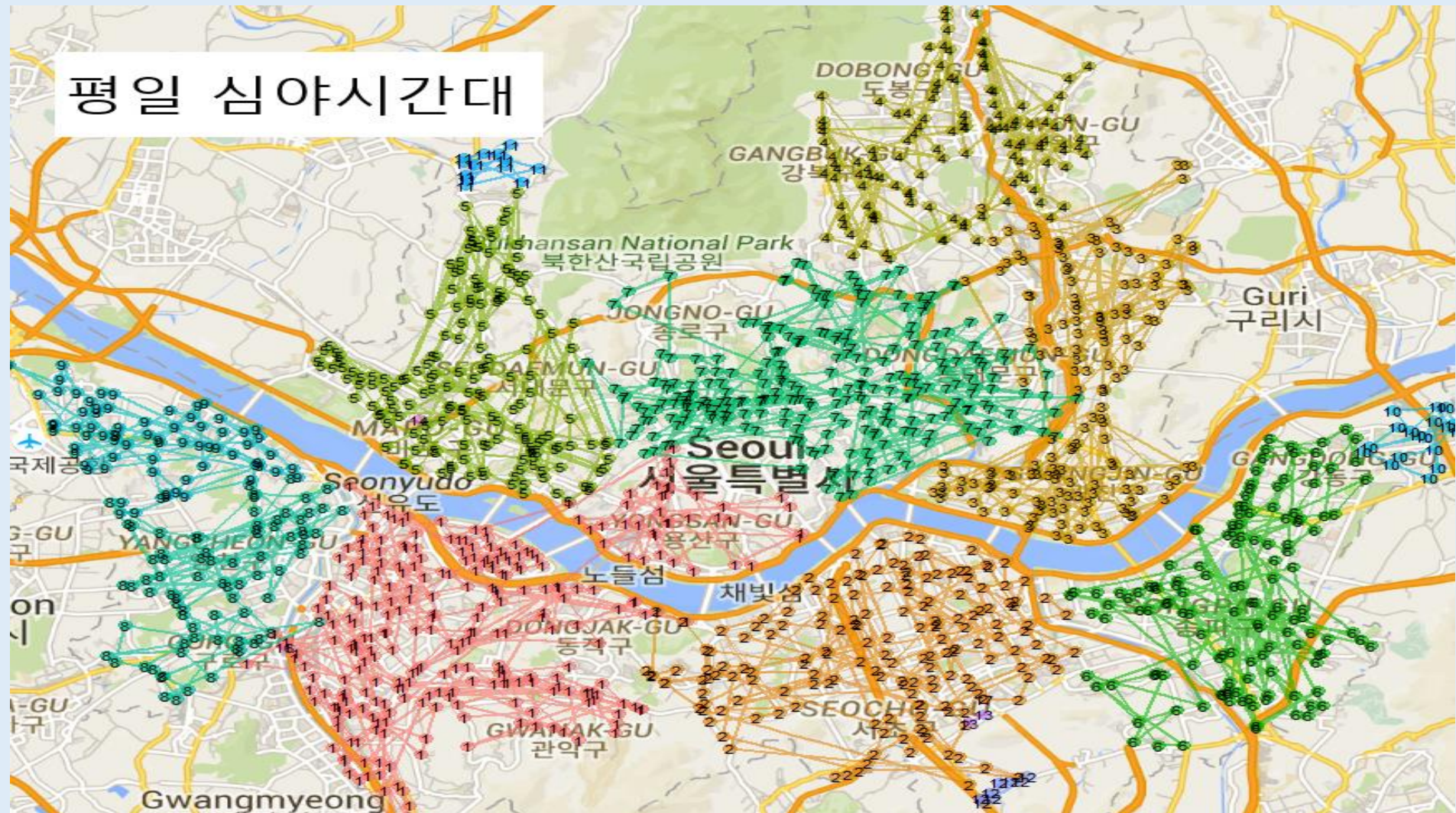




## 02modeling 결과 시각화 \_네트워크 분석

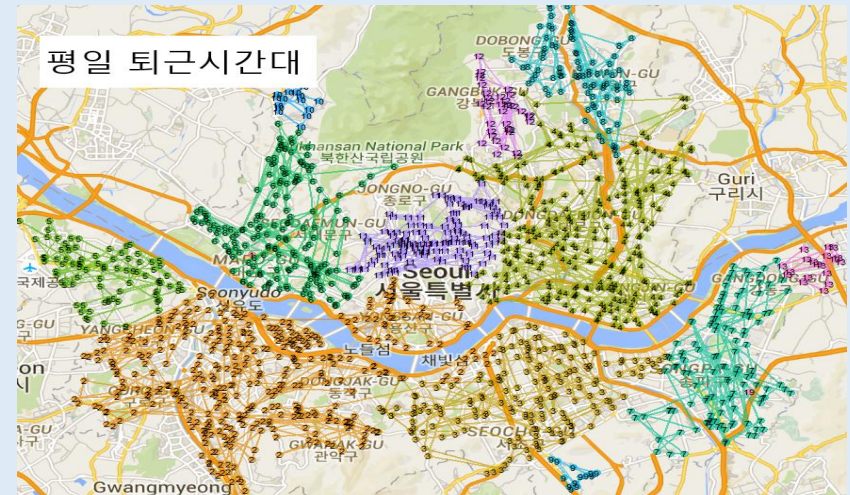
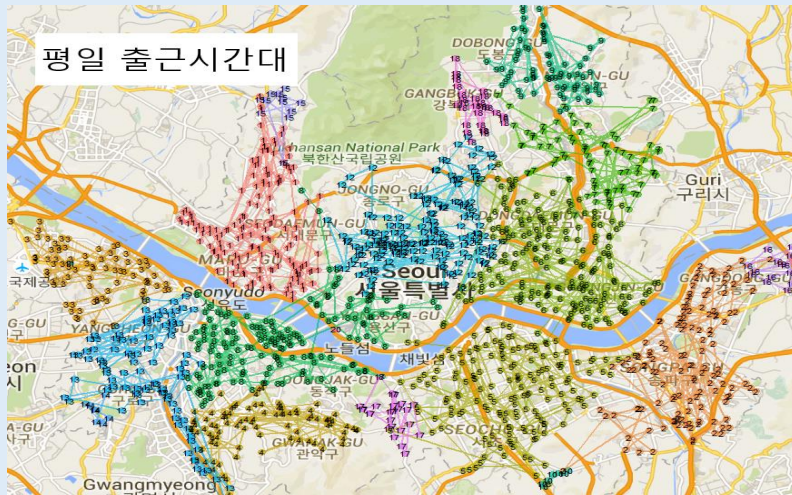
### 평일\_네트워크 분석 결과

평일 심야시간대

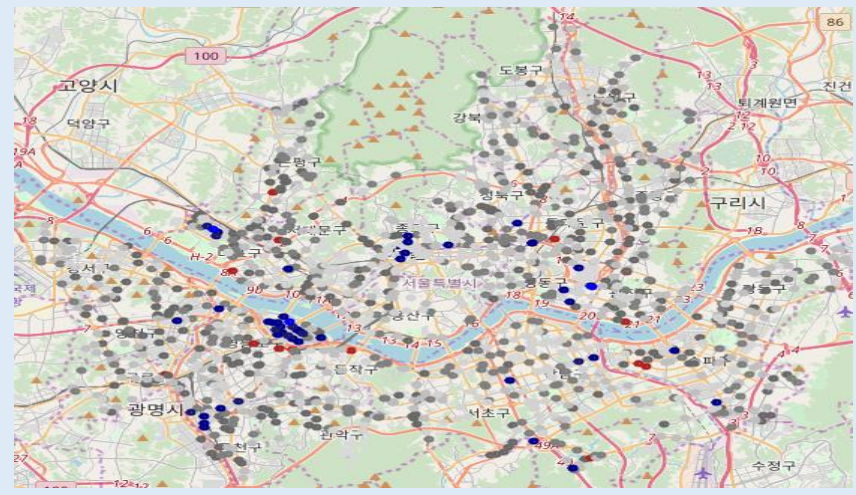
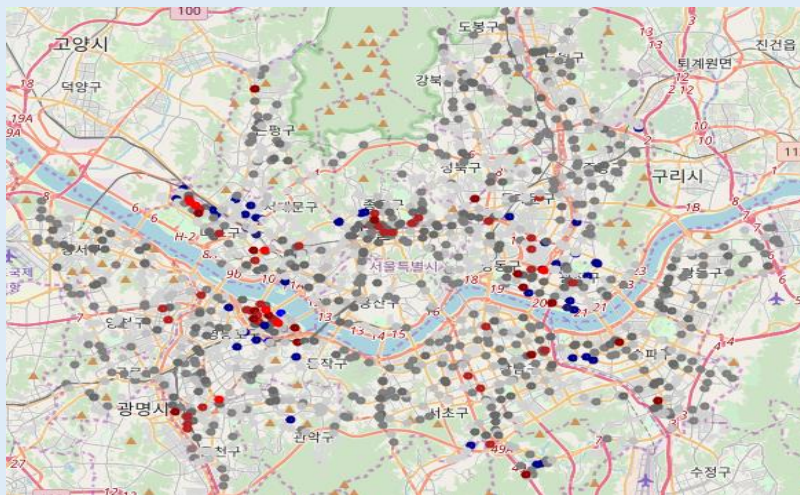




# 02modeling 결과 시각화 -네트워크 분석



# (대여-반납)로 만든 시각화, 빨강: 자전거 쌓임 / 파랑: 자전거 부족



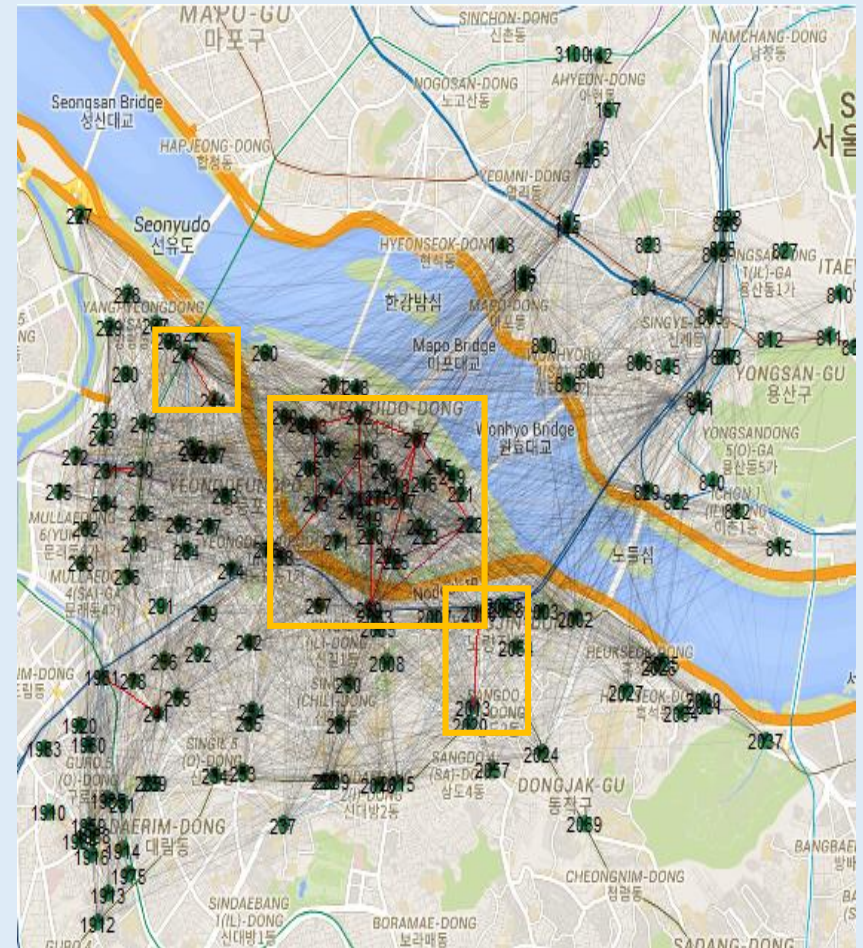
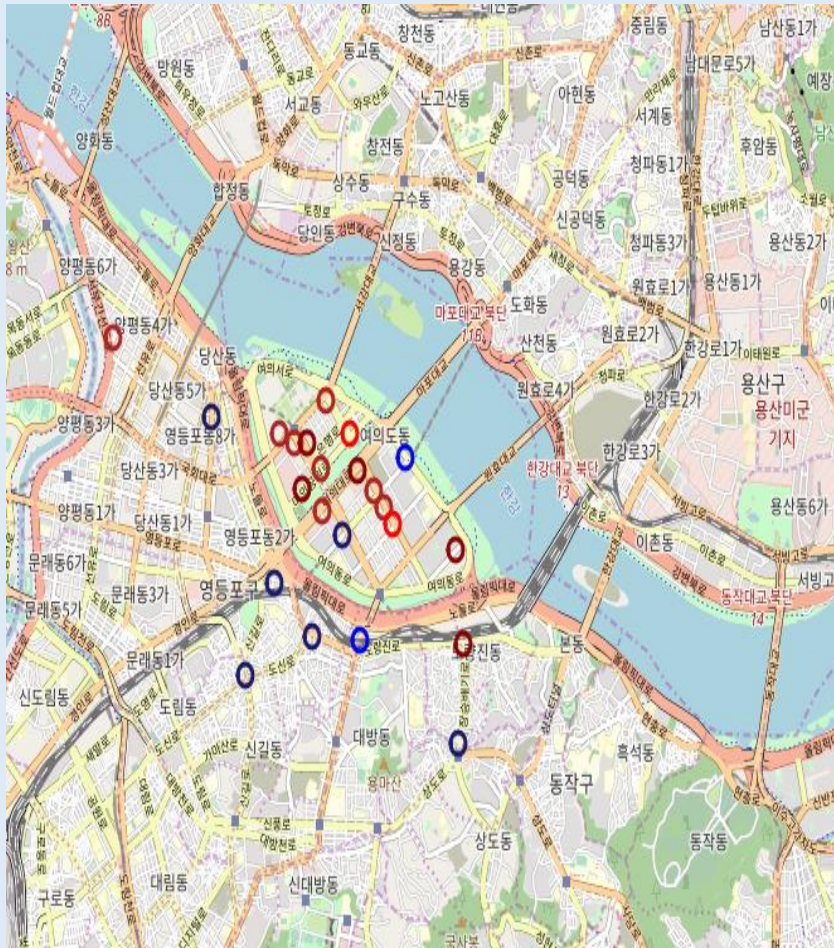
한 네트워크 안에 대여와 반납의 편중이 생김을 확인할 수 있다.



# 02modeling 결과 시각화 - 네트워크 분석

Ex) 여의도 출근

- ✓ (그래프 1) 빨강: 자전거 쌓임 / 파랑: 자전거 부족
- ✓ (그래프 2) 빨강(이동 400 이상)

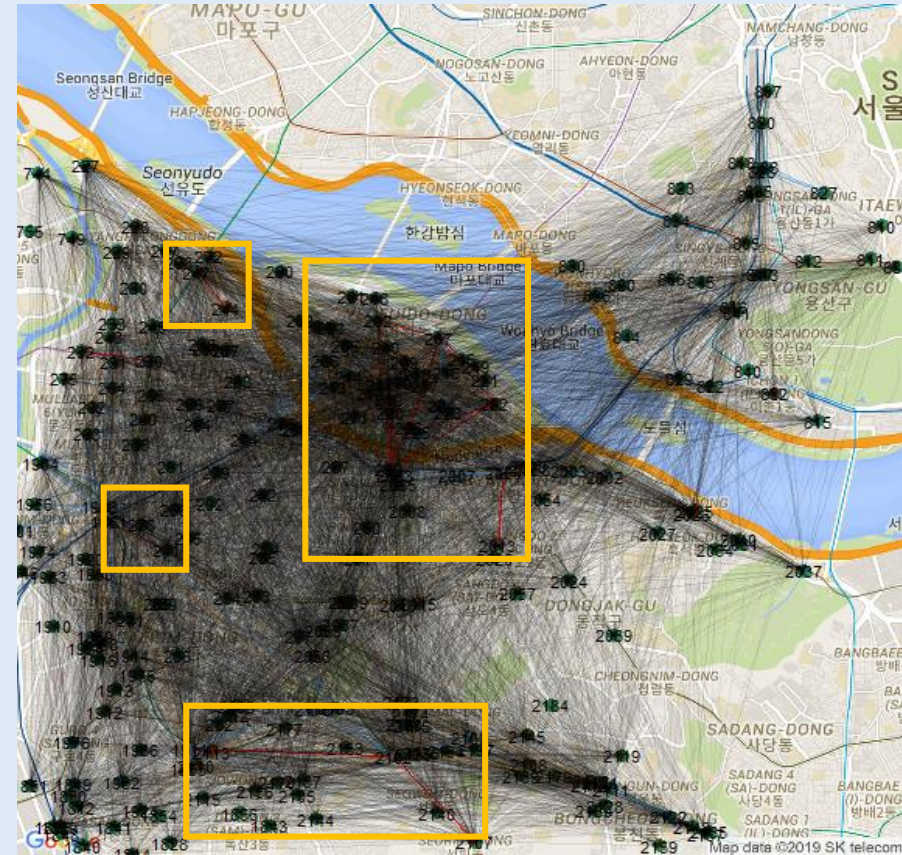
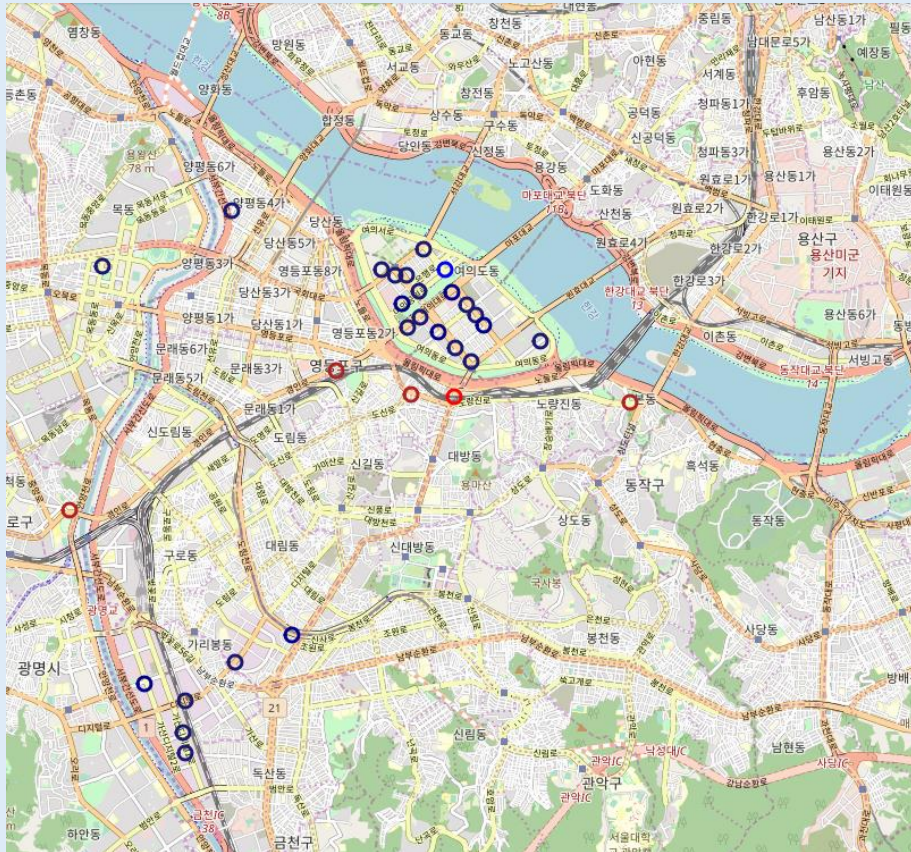




# 02modeling 결과 시각화 -네트워크 분석

Ex) 여의도 퇴근

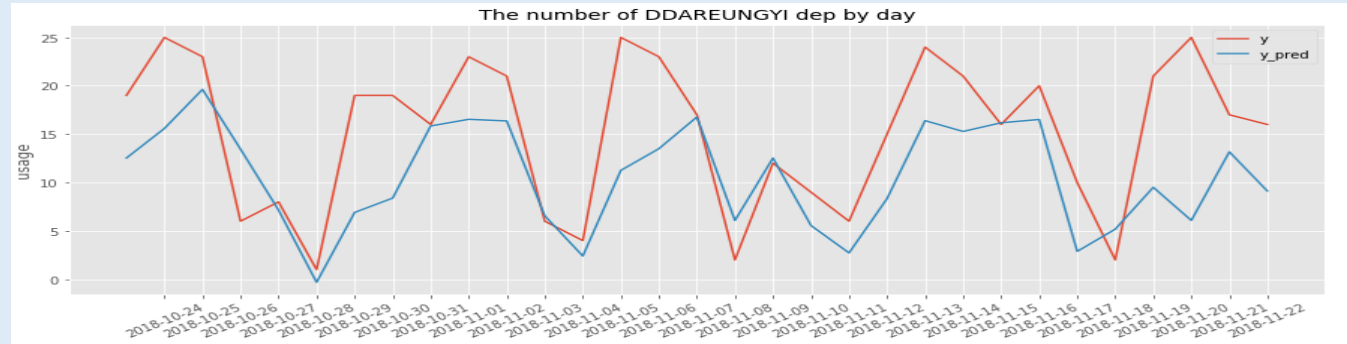
- ✓ (그래프 1) 빨강: 자전거 쌓임 / 파랑: 자전거 부족
- ✓ (그래프 2) 빨강(이동 400 이상)



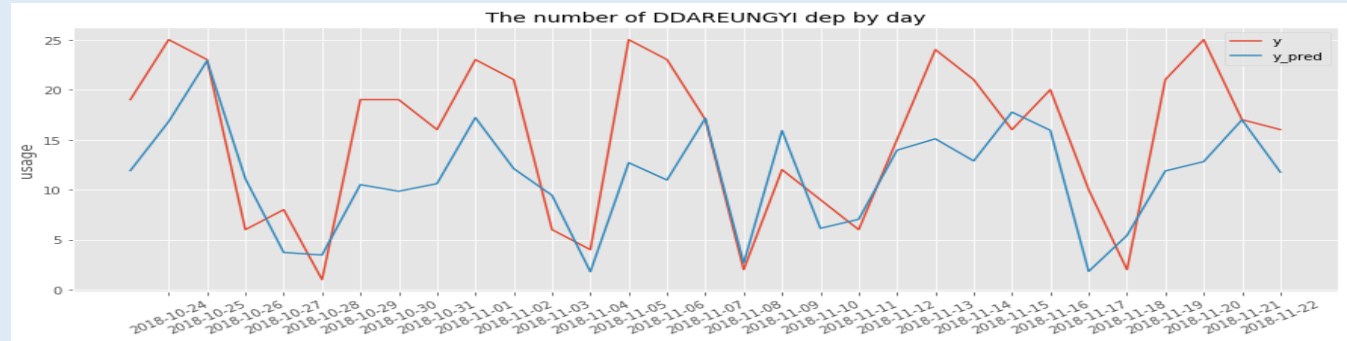
# 02modeling 결과 시각화 \_수요예측분석

## 113번 대여소\_ 대여가 제일 많은 곳

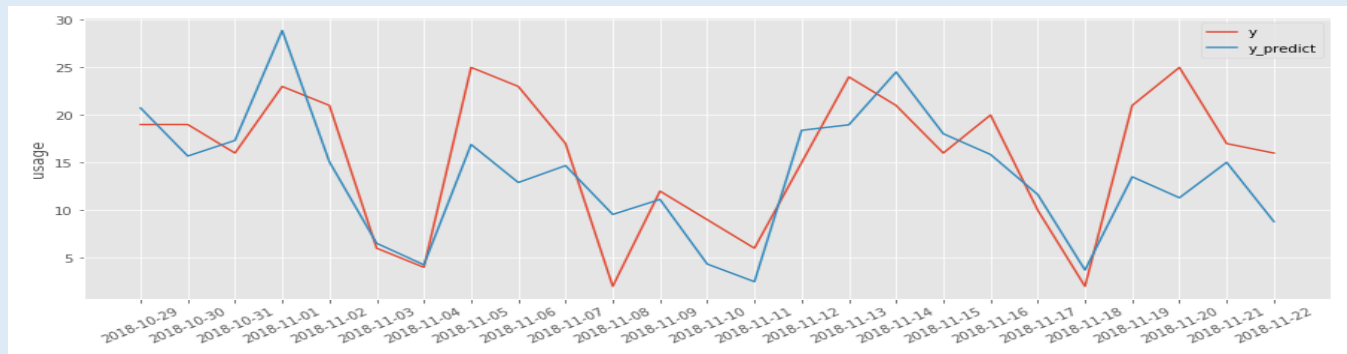
### 1. 깊은 MLP (multilayerperceptron )



### 2. 깊은 MLP (multilayerperceptron ) poly = PolynomialFeatures(degree=2)



### 3. multivariate LSTM time series data prediction



03

**결론**  
**(적용방안)**

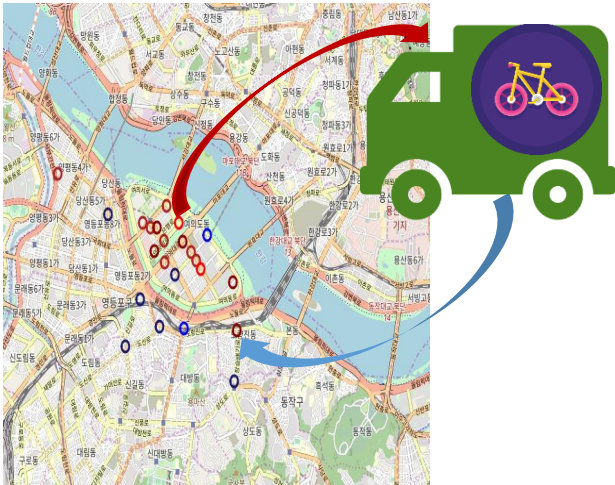
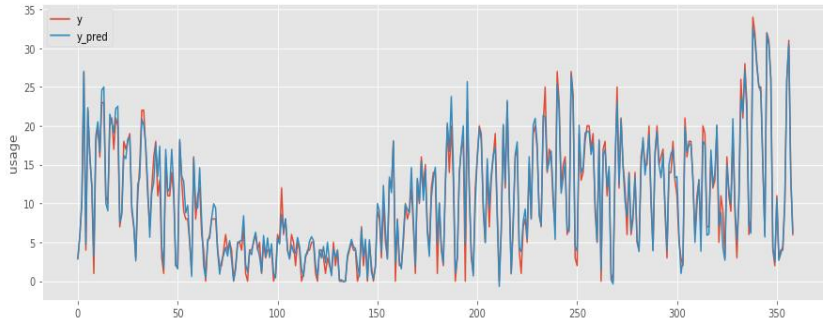


# 결론

## 편중 해소를 위한

## 한 발 빠른 따릉이 분배작업

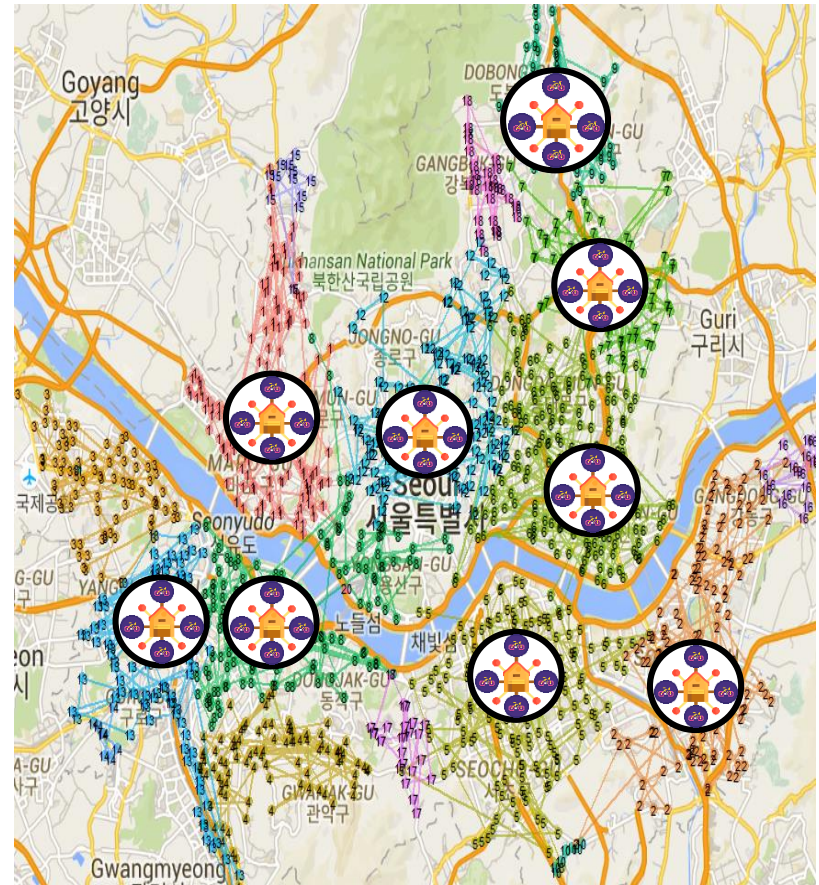
네트워크 내 대여/반납의 편중이 큰 대여소에  
대여와 반납 예측을 한다면,  
반납이 많은 대여소에서 대여가 많은 대여소로  
분배작업의 동선을 선정할 수 있을 것이다.



## 분배 작업을 관리하는

## 권역별 지역 센터 설립

지역구 기반이 아닌 대여/반납의 패턴으로  
권역별 지역 센터의 최적 입지를 선정한다면  
효과적이고 효율적인 분배작업을 수행할 수 있을 것이다.





Q & A



**감사합니다**