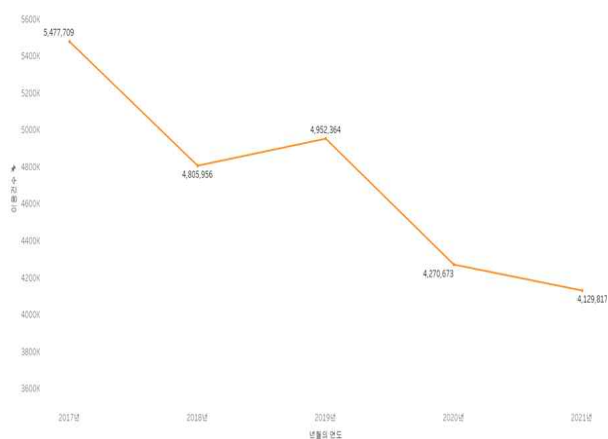


# 2022 창원 빅데이터 공모전 분석보고서

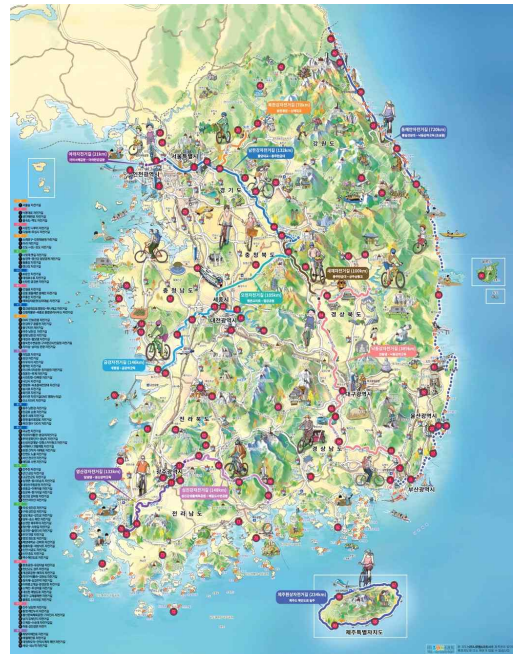
## 주요 누비자 이용 경로 추천과 관광지 자전거 코스 추천

### 1. 분석개요



[그래프 1 - 년도 별 누비자 이용 추이]

[그림1 - 추천 자전거 여행길\_아름다운 자전거길 100선 | 자전거 행복나눔]



지난 5년간, 창원시의 누비자 관련 예산은 증가하거나 유지되는 반면 이용률은 감소하는 추세이다. 이러한 정보는 뉴스 및 기사에서도 쉽게 접할 수 있을 만큼 체감되는 사실이다. 민영 자전거, 전동 킥보드, 코로나 사태 등등 여러 요인이 있겠지만, 창원시가 누비자에 지속적인 노력을 들일 계획이라는 기사를 보며 효율적이고 효과적인 대안을 위해 뿌리가 될 수 있는 조사와 분석이 필요하다 판단하였다. 누비자의 현황 조사뿐만 아니라 누비자 관련 데이터를 분석하여 이용 시간대, 이용 시간(대여 시간), 경로 등을 통해 인사이트를 도출하고자 한다. 나아가 최근 창원시 교통사고 통계 및 뉴스를 보면 자전거 사고로 인해 사망자가 발생하였고, 그 원인 중 하나는 관리 소홀이라는 점이다. 이는 많은 이들로부터 질타를 받는 사항이다. 자전거 사고는 오늘날까지 심심치 않게 들리는 소식이기 때문에 우리는 사람들이 주로 많이 이용하는 터미널과 도착하는 터미널을 분석하며 이동 경로를 예측함으로써 창원시에서 진행하는 관리 감독에 도움이 되고자 한다.

누비자를 조사하던 중에 아래와 같은 사진을 확인할 수 있었다. 위 지도는 전국 자전거 지도, 도로 등 검색할 때 손쉽게 볼 수 있는 지도이다. 놀랍게도, 가까운 울산, 경주, 부산 등은 포함되어 있지만 창원시에 속하는 5개의 행정구 모두 위의 지도에 포함되어 있지 않은 것을 알 수 있었다. 문화관광 도시로서 더욱 발돋움하기 위해 관광객 유치와 아이디어 및 관리 등에 힘쓰고 있는 창원시 입장에서 안타까운 현황이다. 누비자 추천 도로(테마 코스)는 2가지에 불과하며 많은 홍보가 부족한 상황이다. 연합뉴스에 따르면 코로나 바이러스 방역 조치 완화 이후 전북 전주시를 찾는 관광객이 늘면서 공영 자전거 이용객도 급증한 것으로 나타났다고 한다.

#### 참고자료 4.C.

전주시는 올해 상반기의 공영 자전거 대여 건수는 9천713건으로 작년 같은 기간 6천719건보다 2천994건(44.6%) 늘었다고 밝혔다. 한옥마을을 비롯한 전주시 주요 명소를 찾는 관광객이 증가한 데다 공영 자전거를 전반적으로 발전 및 관리하였기 때문으로 해석된다. 창원시의 경우 주남저수지, 내수면환경생태공원, 해양드라마세트장 등 다양한 관광지가 존재하며 빅데이터 포털에 기재된 연도별 관광객 수를 보면 매년 증가 추세를 확인할 수 있다. 또한, 최근 동부마을 팽나무, 해양드라마세트장의 경우 유명 드라마 촬영지로 이용객 증가를 기대할 수 있는 상황이다.

#### 참고자료 4.D.

따라서 우리는 상기 내용과 관련된 데이터를 분석하여 누비자 일반적인 현황 분석뿐만 아니라 터미널 정보 및 경로 내용을 통해 인사이트를 도출, 관광지 데이터를 분석하고 누비자 데이터와 결합하여 새로운 경로를 추천하고자 한다. 전주시 사례와 같이 누비자 이용객도 늘리고 관광객을 유치하는 것이 목표이다.

## 2. 분석결과 상세내용

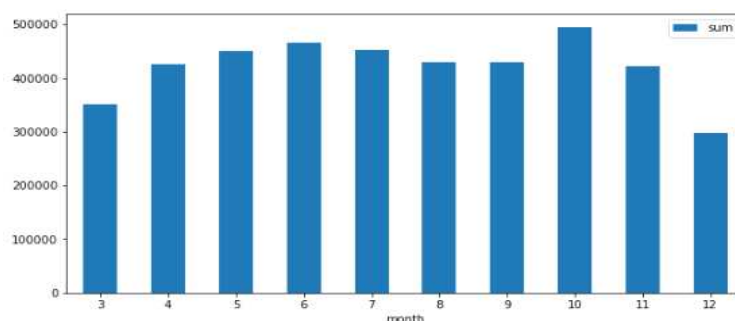
### - 분석에 사용된 데이터 목록 개요

- 누비자 홈페이지 정보 수집: 창원시 빅데이터 포털 사이트에 기재된 누비자 터미널 정보(참고자료 1.C.)가 실제 누비자 홈페이지와 최신화가 되어있지 않아 크롤링을 통해 홈페이지의 터미널 정보(참고자료 2.A.)를 수집했다.
- “터미널 정보” 최신화: 기존 터미널 정보와 크롤링한 터미널 정보를 터미널 명 기준으로 병합 및 최신화를 진행하였다.
- “(신)누비자 반납이력”: 최신화된 터미널 정보를 기준으로 공공데이터포털에서 받은 창원레포츠파크\_창원시무인대여공영자전거 (참고자료 1.A.)를 최신 버전으로 업데이트 하였다.
- “path\_count”: (신)누비자 반납이력에서 출발터미널에서 도착터미널로 누비자의 이동 경로를 기준으로 이동 횟수를 추출하였다(path\_count.csv).
- “시간대별 이용대수”: “(신)누비자 반납이력”에서 각 시간대별 이용 대수를 추출하였다.
- “터미널 시간정보”: 시간대별 이동대수에서 각 터미널별 시간정보를 추출하였다.
- “관광지\_주변터미널정보”: 창원시 빅데이터 포털 사이트에 기재된 (참고자료 1.D.), 터미널 위치 시각화를 바탕으로 관광지 주변 터미널을 정리하였다(관광지\_주변터미널.csv).
- 관광지 통계 현황: 창원시 빅데이터 포털 사이트에 개시된 자료 참고하였다.

※ 분석툴: Jupyter, Python3.0, Pandas, Beautifulsoup, Folium, Tableau

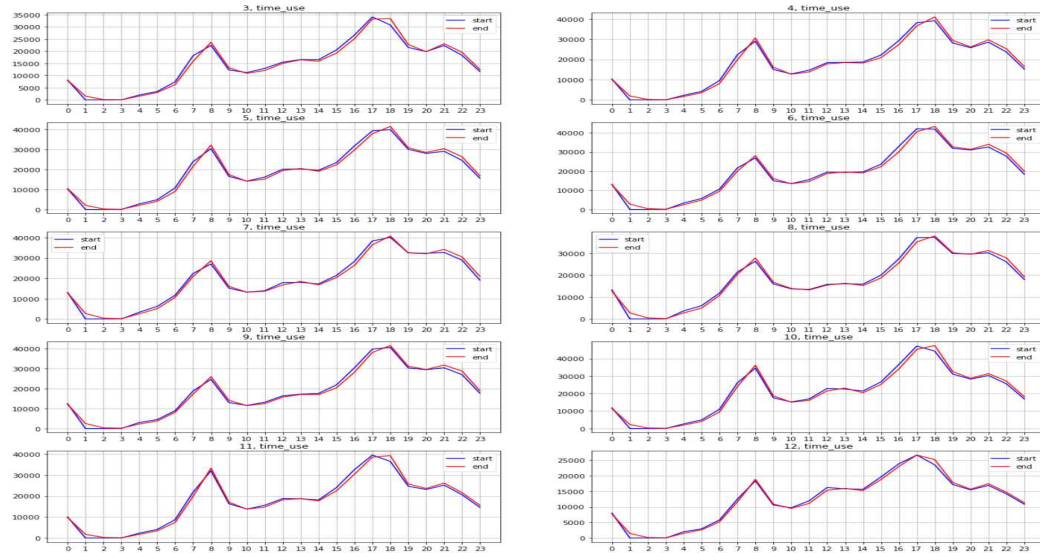
### - EDA

#### 1) 누비자 자전거 현황

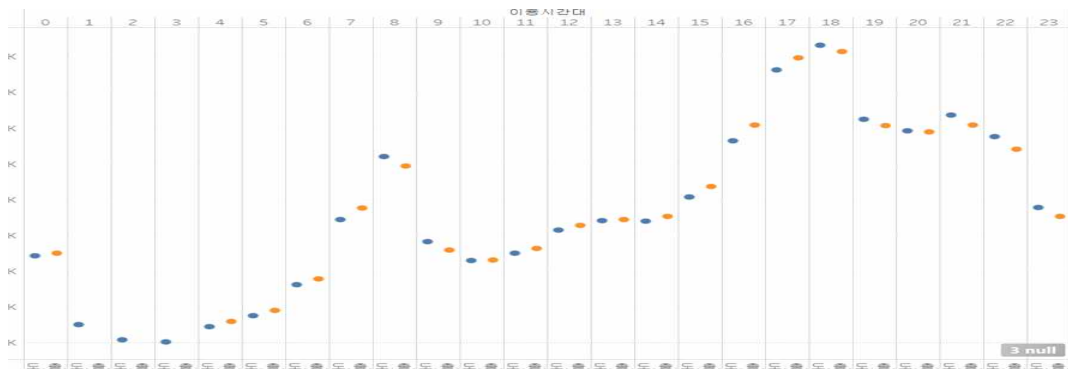


[그래프 2- 월별 누비자 사용현황]

위의 과정으로 인해 생성한 터미널 정보 최신파 파일((신)누비자 반납이력)의 대여 일자와 대여 시간 데이터 열을 병합하여 datetime 형식의 열을 생성하였다. 해당 열을 문자열로 변환시킨 후 일, 시, 분, 초에서 시만 추출하여 각 시간대의 개수를 확인한 후, 시간에 따른 이용량 데이터를 추출하였다.

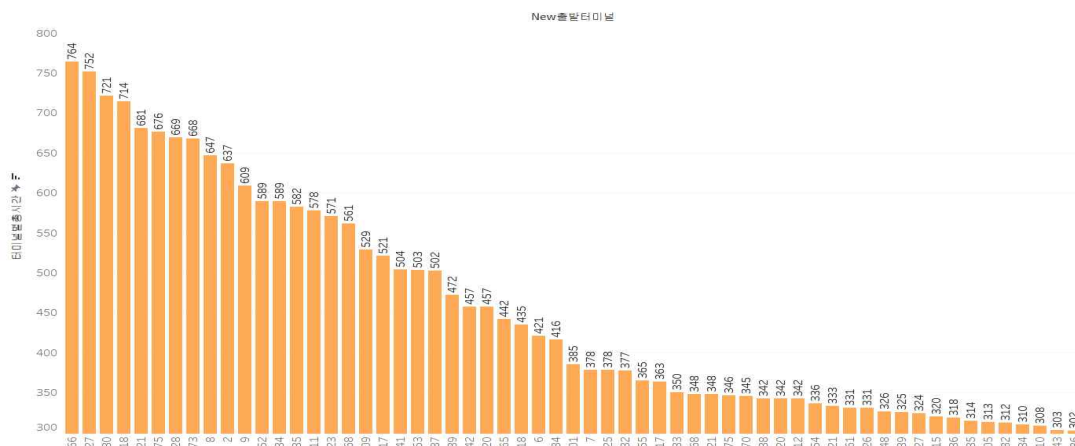


[그래프3 - 월별 누비자 이용 시간대 데이터]



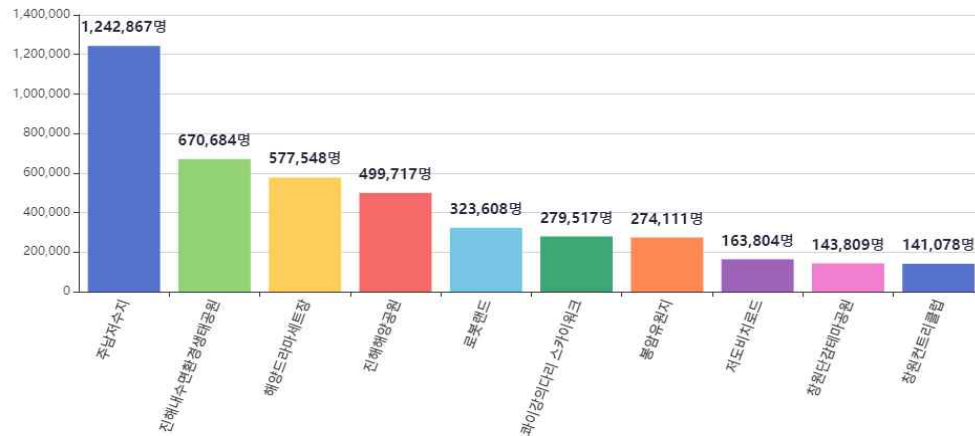
[그래프4 - 전체 년도 누비자 이용 시간대 데이터]

대여 일자와 시간, 반납 일자와 시간 각각을 결합하여 date 형식의 열을 2개 생성하였다. 각 시각을 빼서 이용 길이를 도출하였다. 출발터미널의 이용길이를 모두 합한 값을 추출 후 데이터 프레임으로 만들었다.



[그래프5 - 터미널별 누비자 사용 시간]

## 2) 관광지 현황



[그래프6 - 2021년 창원시 누적 관광객]

[표1 - 이용객 수가 집계되지 않아 도표에 포함되지 않은 관광지]

마금산온천	마산해양레포츠센터	웅천도요지전시관	진해해양레포츠센터	마산 오동동 통솔골목
가고파 꼬부랑길 벽화마을	창동 예술촌	메타 세콰이어 가로수길	굿데이 뮤지엄	여좌천 로망스다리
창원국제사격장	진해드림로드	진해드림파크	경화시장	창원 편백 치유의 숲

주남저수지가 압도적으로 이용객이 많은 것을 확인할 수 있고, 그 뒤로 진해 해양 내수면 환경 생태공원, 해양 드라마 세트장, 진해 해양 공원, 로봇랜드 순으로 관광객이 많은 것을 확인할 수 있다.



[그래프7 - 계절별 관광지 이용객 도표]

계절에 상관없이 주남저수지의 이용객이 가장 많고, 여름에 진해 해양 공원의 이용객이 급격히 올라갔다, 가을에는 창원 단감 테마 공원의 이용객이 증가하였다.

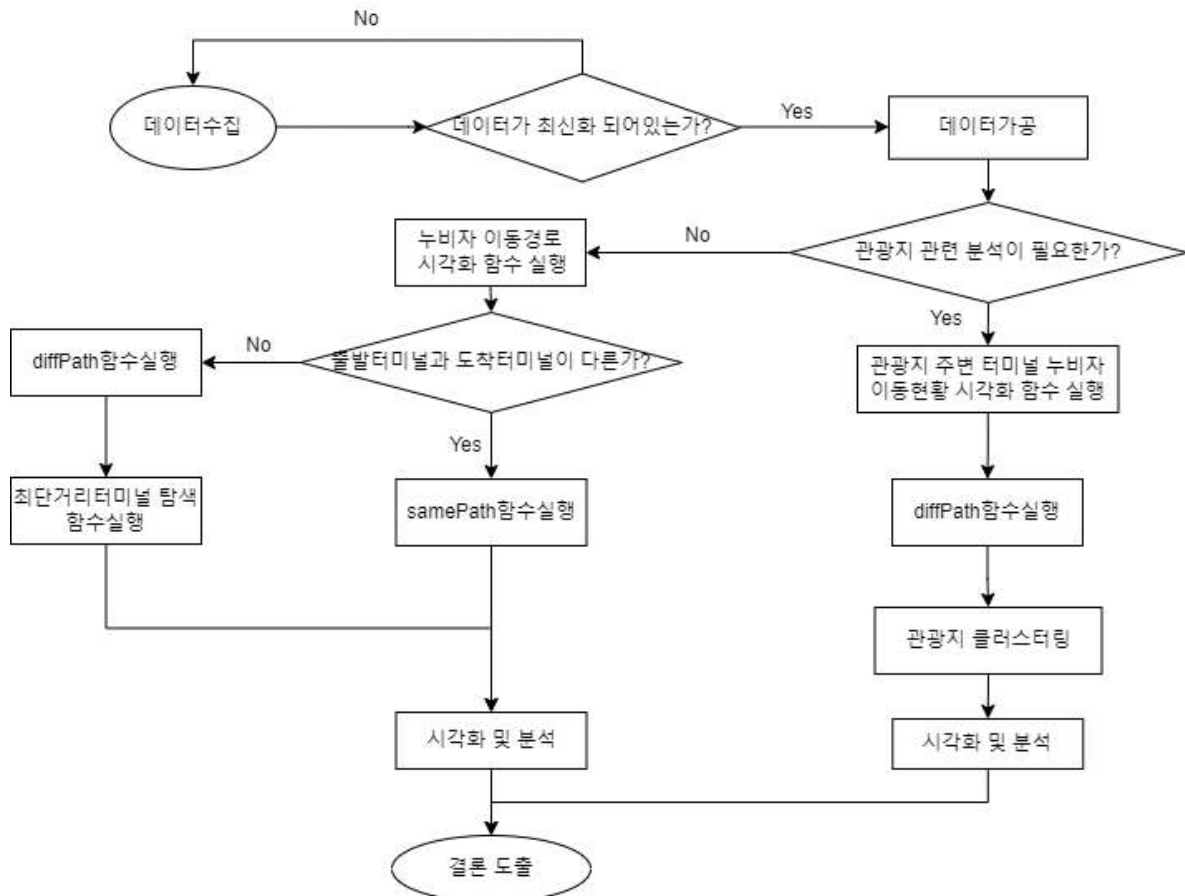
위의 과정으로 인해 생성한 터미널 정보 최신화 파일(누비자 반납 이력)의 출발, 도착터미널 정보를 이용하여 경로 분석을 계획하였다. 다음과 같은 전처리 및 분석 과정을 통해 결론을 도출할 수 있었다. 먼저 새롭게 생성한 터미널 정보 최신화 파일을 생성하는 과정에서 터미널 명을 출발, 도착터미널 번호에 겹쳐치가 존재하였기 때문에, 데이터 프레임에서 해당 행을 제거하였다. 출발, 도착터미널 번호 사이에 '~' 추가한 새로운 열을 생성, 모든 출발터미널~도착터미널 각 개수를 파악 후 새로운 데이터 프레임을 생성하였다. 마지막으로 새롭게 생성한 열에서 출발, 도착터미널 번호를 추출하여 각각 열을 생성하여 아래와 같은 파일을 생성하였다.

	A	B	C	D
1	경로	new 출발	new 도착	개수
2	222~222	222	222	861
3	18~27	18	27	819
4	18~18	18	18	785
5	117~117	117	117	729
6	27~27	27	27	658
7	2~2	2	2	637
8	9~9	9	9	627
9	8~8	8	8	624
10	27~18	27	18	606
11	34~34	34	34	597
12	39~39	39	39	585
13	118~222	118	222	559
14	28~28	28	28	556
15	34~35	34	35	552
16	95~95	95	95	546
17	168~168	168	168	543
18	27~263	27	263	543
19	222~118	222	118	535
20	263~263	263	263	533

[그림2 - 2018년 3월 출발, 도착터미널 경로 개수]

월별로 위의 과정을 거쳐 월별 출발, 도착터미널 번호를 기반으로 한 직선 경로의 개수를 추출하였다. 이를 생성된 데이터를 중심으로 아래의 과정을 거쳐 지도에 시각화에 활용할 것이다.

## - 알고리즘 및 방법론



[그림3 - MapEDA 알고리즘]



```
df_merge = pd.read_csv('data/데이터병합.csv', sep=',', encoding='cp949')
df_merge.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 283 entries, 0 to 282
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   터미널명    283 non-null    object
1   주소        282 non-null    object
2   등록일자    282 non-null    object
3   위도        282 non-null    float64
4   경도        282 non-null    float64
5   번호        283 non-null    int64
6   보관대수    283 non-null    int64
dtypes: float64(2), int64(2), object(3)
memory usage: 15.6+ KB
```

[그림 4 - 데이터병합 데이터]

```
df_attraction_terminal = pd.read_csv('data/관광지주변터미널.csv', sep=',', encoding='cp949')
df_attraction_terminal.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61 entries, 0 to 60
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Unnamed: 0  61 non-null    int64
1   터미널명    52 non-null    object
2   주소        52 non-null    object
3   위도        61 non-null    float64
4   경도        61 non-null    float64
5   보관대수    52 non-null    float64
6   관광지      61 non-null    object
7   터미널번호  52 non-null    float64
8   주변 터미널 유무  61 non-null    bool
9   주변관광지  11 non-null    object
10  가볼만한장소  8 non-null    object
11  출발터미널  0 non-null     float64
12  도착대수    0 non-null     float64
13  출발대수    0 non-null     float64
dtypes: bool(1), float64(7), int64(1), object(5)
memory usage: 6.4+ KB
```

[그림 5 - 관광지 주변 터미널 데이터]

```
df_attraction = pd.read_csv('data/관광객통계.csv', sep=',', encoding='cp949')
df_attraction.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 21 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Unnamed: 0  39 non-null    int64
1   관광지      39 non-null    object
2   할계_x      23 non-null    float64
3   2021_01     23 non-null    float64
4   2021_02     23 non-null    float64
5   2021_03     23 non-null    float64
6   2021_04     23 non-null    float64
7   2021_05     23 non-null    float64
8   2021_06     23 non-null    float64
9   2021_07     23 non-null    float64
10  2021_08     23 non-null    float64
11  2021_09     23 non-null    float64
12  2021_10     23 non-null    float64
13  2021_11     23 non-null    float64
14  2021_12     23 non-null    float64
15  할계_y      23 non-null    object
16  2022_01     23 non-null    float64
17  2022_02     23 non-null    float64
18  2022_03     23 non-null    float64
19  위도        38 non-null    float64
20  경도        38 non-null    float64
dtypes: float64(18), int64(1), object(2)
memory usage: 6.5+ KB
```

[그림 6 - 관광지 정보 데이터]

```
for i in range(3, 13):
    globals()['df_path{}'.format(i)] = pd.read_excel('data/path/path_count{}.xlsx'.format(i), engine='openpyxl')
```

[그림 7 - 각 월별 이동경로 개수 데이터]

“데이터병합” 데이터와 월별 이동 경로 개수 데이터를 받아온다. 이동개수가 많은 상위 데이터만을 확인하기 위해서 이동 경로의 개수가 400개 이상인 데이터만 추출한다. 이후에 병합데이터에서 결측치를 제거한다. 병합데이터에서 필요한 정보인 터미널 번호와 위도, 경도의 데이터를 추출하여 딕셔너리 형태로 터미널 명에 해당 터미널의 위도, 경도를 뽑을 수 있도록 만들었다. 또한 “samePath”함수를 사용하여 출발터미널과 도착터미널이 같은 경우(이하 같은 경로)를 추출하고 “diffPath”함수를 사용하여 출발터미널과 도착터미널이 다른 경우(이하 다른 경로)를 추출하는 같은 경로일 때 주변 터미널의 이동 경로를 시각화하기 위해 가장 가까운 터미널을 탐색, 해당 터미널과의 거리를 계산해주는 “searchNode”함수(이하 최단 거리 터미널 탐색 함수)로 딕셔너리를 제작한다.

지도 시각화 라이브러리 Folium을 활용하여 시각화를 시행한다. 중심 좌표는 창원시청으로 지정하고 창원이 한눈에 보이도록 줌을 설정했다. 이후 추출한 정보들을 활용하여 터미널 위치(Maker)를 시각화를 하였는데, 만약에 같은 경로가 있다면 최단 터미널과의 거리에 0.8배 만큼을 반경으로 하는 원을 지도에 표현하였다. 그리고 다른 경로는 Folium의 “PolyLine”을 활용하여 경로의 개수마다 가중치를 다르게 주어 경로 간의 직선거리를 지도에 표현하였다.

## 누비자 이동경로 시각화함수

```
def mapEDA_path(df_merge, df_path):
    df_path = df_path[df_path['가중치'] >= 400]
    df_merge = df_merge.dropna(axis=0)

    m = folium.Map(
        location=[35.2538, 128.6402],
        zoom_start=11
    )

    coords1 = df_merge[['번호', '위도', '경도']]
    marker_cluster1 = MarkerCluster().add_to(m)

    extract_col = df_merge[['번호', '위도', '경도']]

    terminal_dict = dict()

    for num, lat, long in zip(extract_col['번호'], extract_col['위도'], extract_col['경도']):
        terminal_dict[num] = [lat, long]

    samepath = samePath(df_path)

    searchNode_dict = searchNode(samepath, terminal_dict)

    marker_cluster3 = MarkerCluster().add_to(m)

    for name, lat, long in zip(coords1['번호'], coords1['위도'], coords1['경도']):
        folium.Marker([lat, long], popup=name, icon=folium.Icon(color="green")).add_to(marker_cluster1)
        if name in samepath:
            folium.Circle([lat, long], radius=searchNode_dict[name] * 800, fill=True).add_to(m)

    diffpath = diffPath(df_path)

    for depart_num, arrive_num, weight in diffpath:
        folium.PolyLine(
            locations=[terminal_dict[depart_num], terminal_dict[arrive_num]],
            tooltip='{}->{}, 가중치: {}'.format(depart_num, arrive_num, weight),
            weight=weight,
        ).add_to(m)

    return m
```

[그림 8 - 누비자 이동 경로 시각화 함수]

앞선 데이터들과 관광지 주변 터미널 정보를 받아와서 각 데이터에서 결측치를 제거하였다. 관광지 정보 데이터에서는 관광지명, 위도, 경도 정보, 관광지 주변 터미널 정보에서는 관광지 주변 터미널 번호를 추출하였다. “데이터병합” 데이터에서는 터미널 명, 위도, 경도를 추출하였다.

또한, “데이터병합”에서 필요한 정보인 터미널 번호와 위도, 경도의 데이터를 딕셔너리 형태로 터미널명을 통해 해당 터미널의 위도, 경도의 정보를 읽을 수 있도록 제작하였다. 관광지 주변 터미널로 오는 다양한 경로를 보기 위해 “월별 이동경로 개수 데이터”에서 “누비자 이동경로 시각화” 함수와는 다르게, 이동 경로 개수가 200개 이상인 데이터를 선별하였다. 이후에는 “diffPath”함수를 사용하여 다른 경로를 추출하였다.

마지막으로 Folium을 활용, 지도 시각화를 실시한다. 중심 좌표는 앞과 동일하게 창원시청으로 설정하고, 추출한 정보들을 활용하여 터미널 위치(Maker)와 관광지 위치(Maker)를 지정하였다. 관광지의 경우 주변 반경 1km를 원 형태로 표현하였다. 또한 Folium의 PolyLine을 활용하여 관광지 주변 터미널로 향하는 경로만 필터링하여 각 가중치로 굵기를 다르게 하여 직선 경로를 표현하였다.

## 관광지 주변 터미널 누비자 이용현황 시각화함수

```
def mapEDA_attr(df_merge, df_attraction, df_path, df_attraction_terminal):
    df_merge = df_merge.dropna(axis=0)
    df_attraction_terminal = df_attraction_terminal['터미널번호']
    df_attraction = df_attraction[['위도', '경도', '관광지']]
    df_attraction = df_attraction.dropna(axis=0)
    df_attraction_terminal = df_attraction_terminal.dropna(axis=0)
    df_attraction_terminal = df_attraction_terminal.astype('int')
    extract_col = df_merge[['번호', '위도', '경도']]

    terminal_dict = dict()

    for num, lat, long in zip(extract_col['번호'], extract_col['위도'], extract_col['경도']):
        terminal_dict[int(num)] = [lat, long]

    df_path = df_path[df_path['가중치'] >= 200]

    diffpath = diffPath(df_path)

    m = folium.Map(
        location=[35.2538, 128.6402],
        zoom_start=11
    )

    coords1 = df_merge[['번호', '위도', '경도']]

    marker_cluster1 = MarkerCluster().add_to(m)

    for name, lat, long in zip(coords1['번호'], coords1['위도'], coords1['경도']):
        folium.Marker([lat, long], popup= [name], icon = folium.Icon(color="green")).add_to(marker_cluster1)

    coords2 = df_attraction[['관광지', '위도', '경도']]

    marker_cluster2 = MarkerCluster().add_to(m)

    for name, lat, long in zip(coords2['관광지'], coords2['위도'], coords2['경도']):
        folium.Marker([lat, long], tooltip= name, icon = folium.Icon(icon="star", color="red")).add_to(marker_cluster2)
        folium.Circle([lat, long], radius = 1000, fill=True, color="red").add_to(m)

    for depart_num, arrive_num, weight in diffpath:
        if arrive_num in list(df_attraction_terminal):
            folium.PolyLine(
                locations = [terminal_dict[depart_num], terminal_dict[arrive_num]],
                tooltip = '{}->{}, 가중치: {}'.format(depart_num, arrive_num, weight),
                weight= weight,
                color = "purple"
            ).add_to(m)

    return m
```

[그림 9 - 관광지 주변 터미널 누비자 이용현황 시각화 함수]

## 다른 경로 추출 함수

```
def diffPath(df_path):
    diffpath = []
    for path, count in zip(df_path['경로'], df_path['개수']):
        if count >= 1100:
            weight = 10
        elif count >= 900:
            weight = 8
        elif count >= 700:
            weight = 6
        elif count >= 500:
            weight = 4
        elif count >= 300:
            weight = 3
        else:
            weight = 1

        sp_lst = path.split('~')
        if sp_lst[0] != sp_lst[1]:
            diffpath.append([int(sp_lst[0]), int(sp_lst[1]), weight])

    return diffpath
```

[그림 10 - diffPath함수]

## 같은 경로 추출 함수

```
def samePath(df_path):
    samepath = []
    for path in df_path['경로']:
        sp_lst = path.split('~')
        if sp_lst[0] == sp_lst[1]:
            samepath.append(int(sp_lst[0]))

    return samepath
```

[그림 11 - samePath함수]

“diffPath함수”(출발터미널과 도착터미널이 다른 경로를 추출하는 함수) 각 경로의 이동회수에 따라 가중치를 다르게 두었다. 이동회수 200을 1로 설정, 200단위로 가중치를 10까지 증가시킨다. 출발터미널과 도착터미널, 가중치를 반환하여 추후 시각화에 활용된다.

“samePath”함수(같은 경로 추출하는 함수)를 통해 출발터미널과 도착터미널이 같은 경로(같은 터미널에서 빌려서 같은 터미널에 반납한 경우)를 추출하였다.



## 최단거리 터미널 탐색 함수

```
def searchNode(samepath, terminal_dict):
    searchNode_dict = dict()
    for stdnum in samepath:
        distance = 100000
        for terminal_num in list(terminal_dict.keys()):
            if terminal_num == stdnum:
                continue
            node1 = terminal_dict[stdnum]
            node2 = terminal_dict[terminal_num]
            dist = haversine(node1, node2)
            if dist < distance:
                distance = dist
                result = [stdnum, terminal_num, distance]
            searchNode_dict[result[0]] = distance
    return searchNode_dict
```

[그림 12 - 최단거리 터미널 탐색 함수]

“데이터병합”에서 필요한 정보인 터미널 번호와 위도, 경도 정보를 추출하여 딕셔너리 형태로 제작한 “terminal\_dict”와 “samePath”함수에서 반환된 samePath 리스트를 매개변수로 받아낸다. 같은 경로의 경우들을 return을 통해 받아내고 이중 for문 이용하여 위도, 경도의 정보를 활용하여 거리를 계산해주는 “haversine”함수를 사용한다. 기준 터미널과 모든 터미널과의 거리를 계산하고 계산할 때 마다 최단터미널과 최단 거리를 최신화하고, 마지막에는 최단 터미널과 최단 거리를 딕셔너리 형태로 반환해준다.

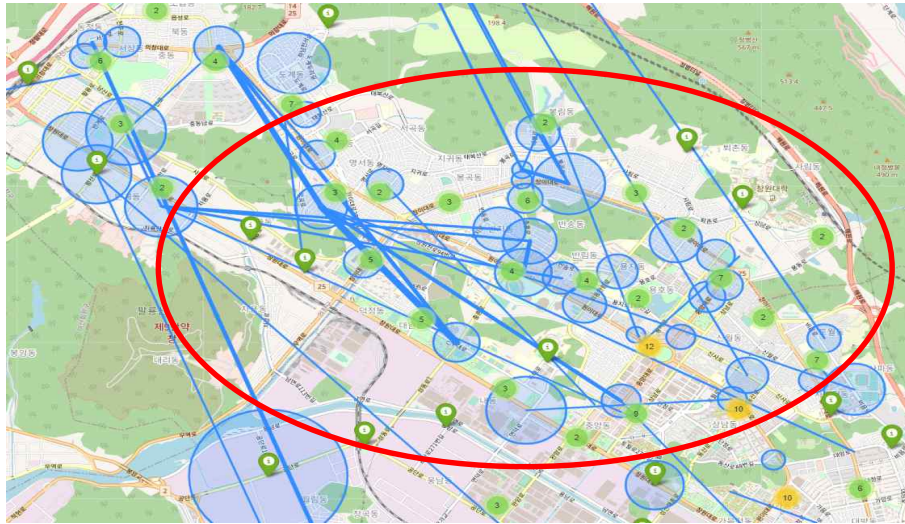
관광지 클러스터링의 경우 다음과 같은 알고리즘을 통해 구현하였다. 우선 주요 관광지 통계자료를 활용하여 상위 10개의 관광지를 정렬하였다. 정렬한 상위 10개를 중심 노드로 선정하고 거리 기반 군집화를 이행하였다. 이때 거리는 직선거리를 우선하며 직선 경로상 바다, 산, 공단이 있다면 가중치를 inf로 두었다. 또한 만일 범위 내에 누비자의 이동 경로가 존재한다면 거리상 경로의 추가적인 이점을 주어 클러스터링을 실시하였다.

### 3. 결과 해석 및 시사점

#### - 누비자 이동 경로 결과 및 정리

월별 누비자 사용현황을 보면 10월의 이용 빈도가 제일 잦다. 따라서 월별로 이동 경로가 다르지만, 10월 누비자 이동 경로를 대표로 선정하여 결과를 도출하였다. 그 외 3~12월까지 누비자 이동 경로를 전반적으로 살펴보면 다음과 같은 내용을 정리하였다.

##### 1) 터미널 밀집도

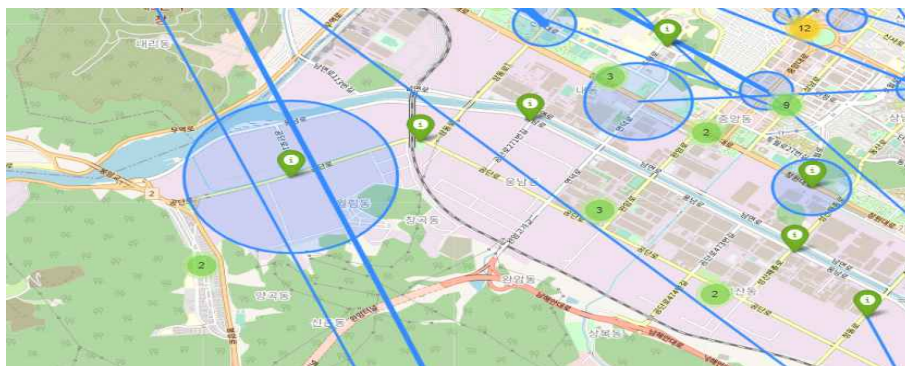


[그림 13 - 누비자 이동 경로 시각화 1]

창원시 전체에서 누비자 이용은 의창구와 성산구가 타 자치구에 비해 압도적으로 많음을 확인할 수 있다. 의창구와 성산구가 타 자치구 대비 인구수가 많은 것이 사실이지만, 타 자치구의 경우 직선 및 원이 없다. 따라서 창원시의 누비자 이용 대부분은 의창구 및 성산구에서 이루어짐을 알 수 있다.

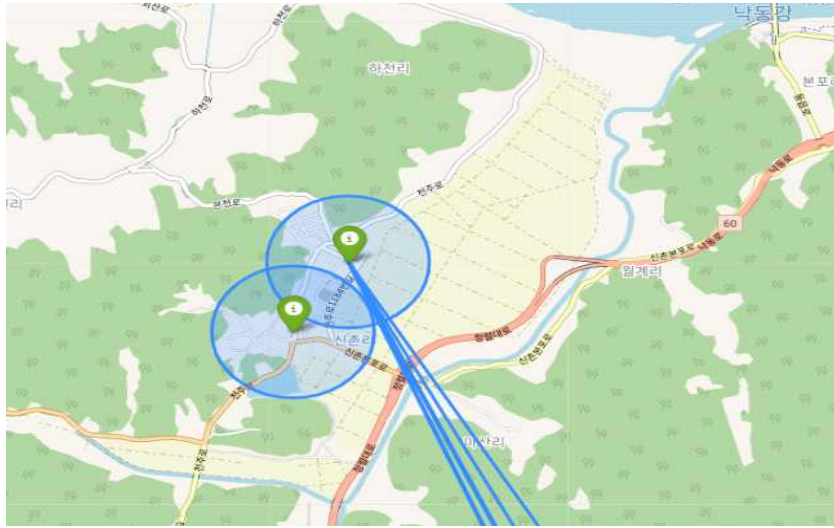
##### 2) 출발터미널과 도착터미널이 같은 경우

원이 그려지는 구역의 경우(출발과 도착 터미널이 같은 경우), 먼저 해당 지역에서 거주민 또는 근로자 등의 이용객이 많다는 것으로 해석이 가능하다. 오랫동안 이용 후에 다시 돌아왔다는 것, 또는 짧게 이용한 후에 다시 돌아왔다는 것, 이러한 상황들은 모두 위의 상황으로 해석이 가능하다. 두 번째는 주변에 공원, 관광지 등이 있어 나들이와 비슷한 목적으로 이용하는 경우가 많다는 것으로도 해석을 하였다.



[그림 14 - 누비자 이동 경로 시각화 2]

월림동 쪽 공단에 위치한 터미널 (신촌 단지 입구 - 267)에 원이 그려지는 것을 볼 수 있다.



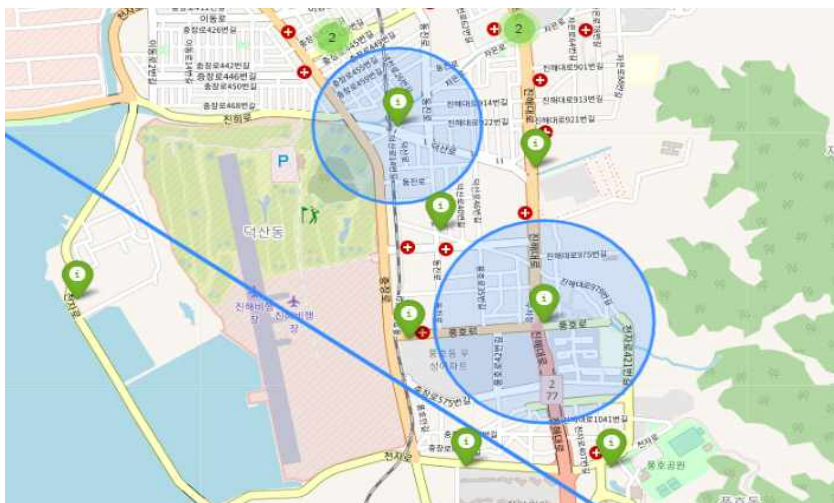
[그림 15 - 누비자 이동 경로 시각화 3]

신촌리 북면에 존재하는 터미널 2개에 모두 원이 그려지는 것을 볼 수 있다.



[그림 16 - 누비자 이동 경로 시각화 4]

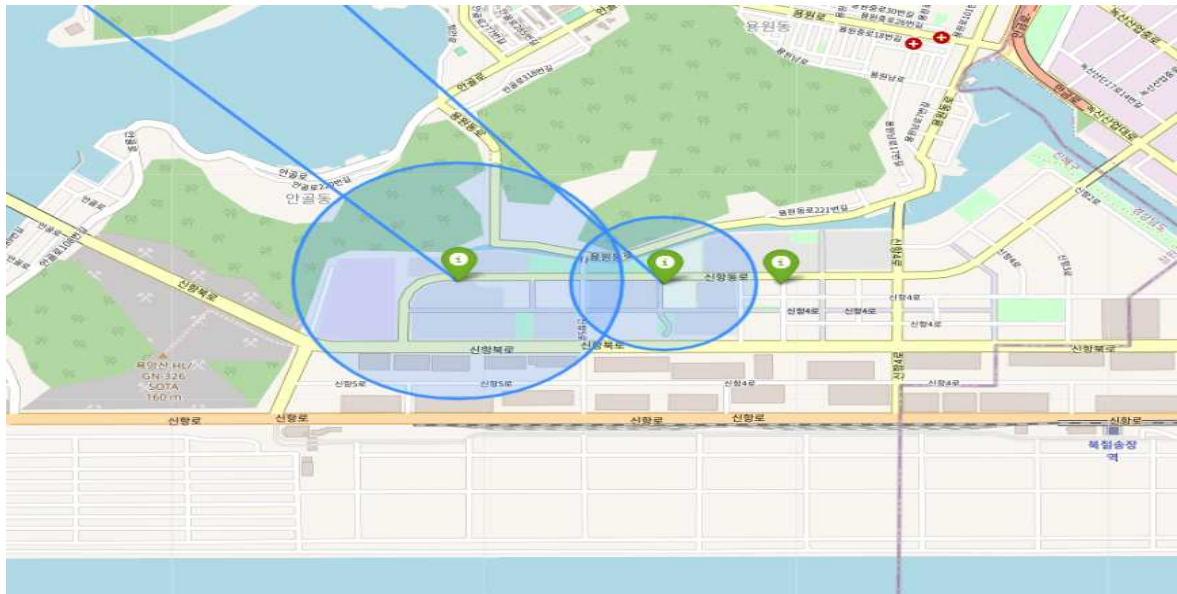
여좌동 여좌천 주변 터미널 2개에 원이 그려지는 것을 볼 수 있다.



[그림 17 - 누비자 이동 경로 시각화 5]



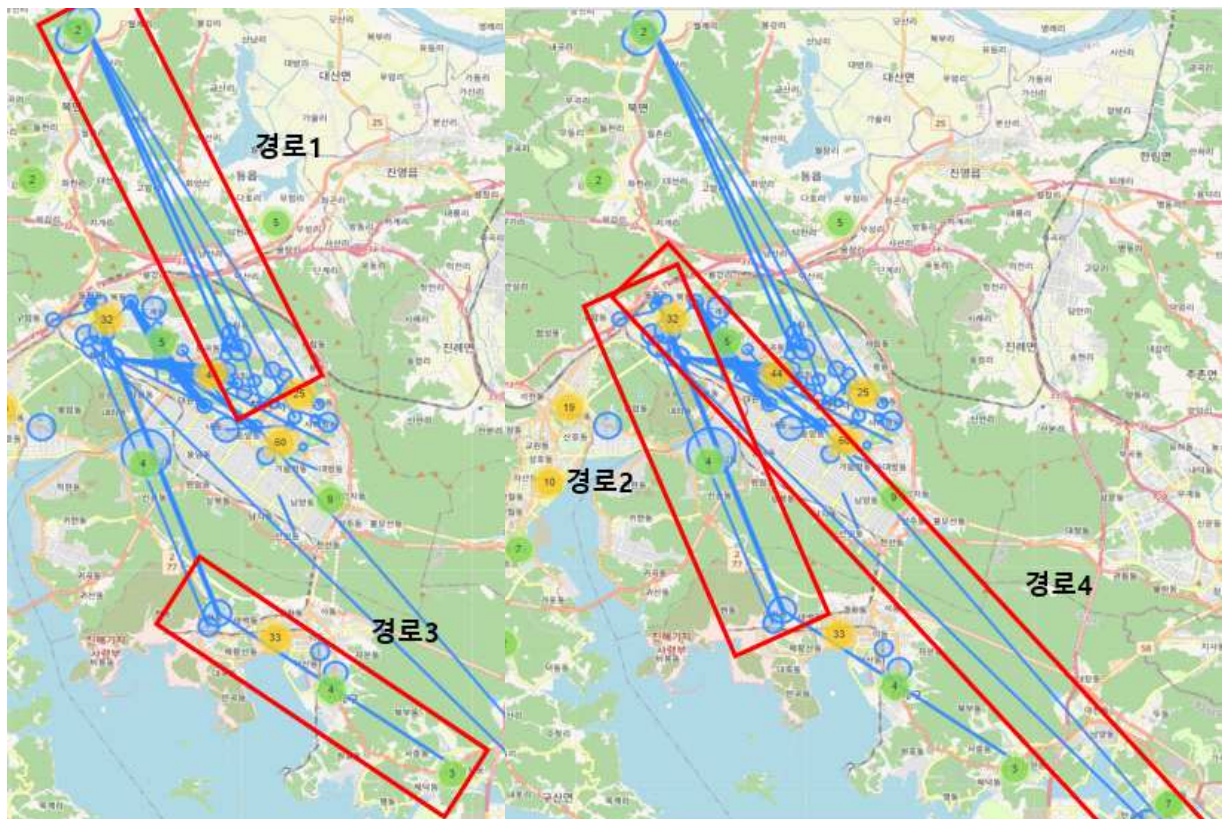
덕산동 쪽 터미널에 (덕산성당 앞 - 232, 풍호초등학교 사거리 - 184) 2개의 원이 그려지는 것을 볼 수 있다. 원이 그려진 지역과 비교 시 직선 경로가 해당 지역을 가로지르거나, 해당 터미널에서 다른 터미널로 이어지는 경로가 보이지 않는다.



[그림 18 - 누비자 이동 경로 시각화 6]

부산 신항 주변 터미널에 (부산 신항 사랑으로 부영3단지입구 - 275, 신항 중학교 - 274) 2개의 원이 그려지는 것을 볼 수 있다.

### 3) 출발터미널과 도착터미널의 거리가 먼 경우 (이하 특이경로라 표현)



[그림 19 - 누비자 이동 경로 시각화 7]

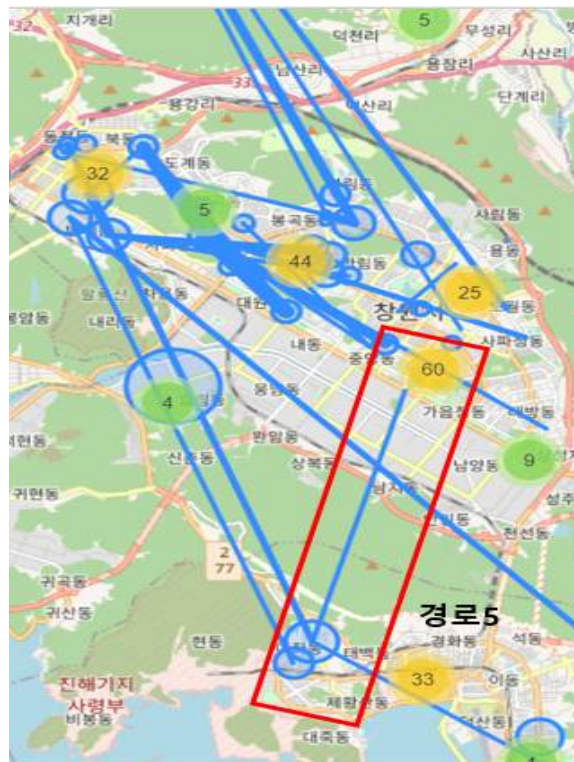


[특이경로 1]: 의창구 및 성산구 주위의 터미널과 북면 신촌리를 오가는 경우에는 가중치가 4이다.

[특이경로 2]: 매달 여좌동 주위의 터미널과 동정동 및 팔용동 주위의 오가는 경우에는 가중치가 6이다.

[특이경로 3]: 여좌동 주위 터미널과 성내동 주위 터미널(웅천동 주민 센터 - 272)을 오가는 경우에는 가중치가 4이다, 하지만, 성내동 웅천동 주민 센터 터미널의 경우 원이 그려지지 않는 점이 특이사항이다.

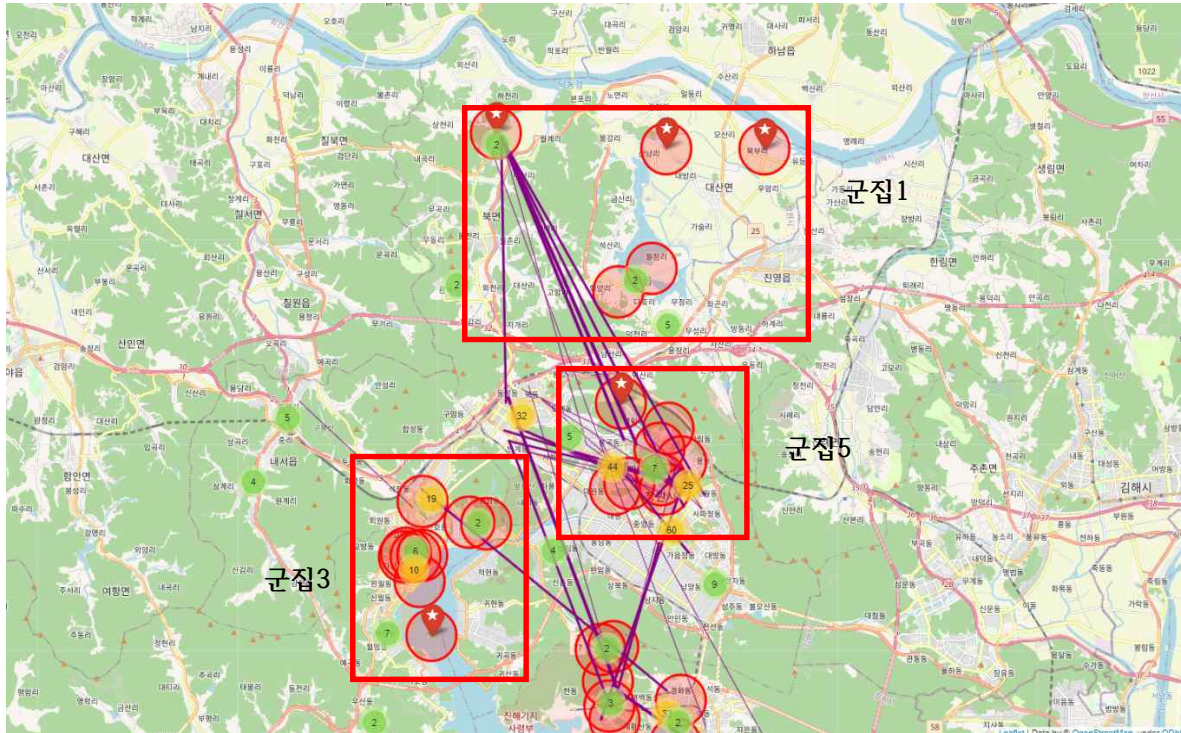
[특이경로 4]: 부산 신항 주위의 터미널과 중앙동, 팔용동 주위의 터미널을 오가는 경우에는 가중치가 3이다. 특이사항으로 거리가 타 특이경로에 비해 멀다.



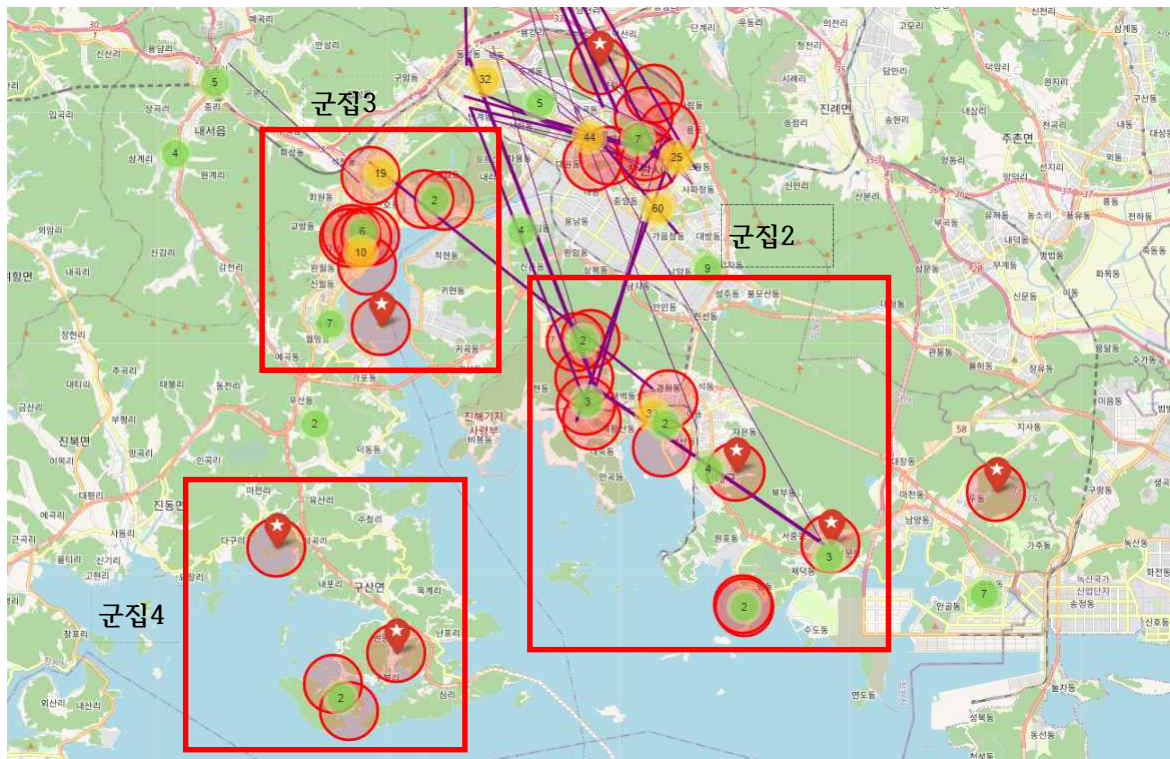
[그림 20 - 누비자 이동 경로 시각화 8]

[특이경로5]: 4~8월의 경우 여좌동 주위의 터미널과 상남동 주위의 터미널을 오가는 경우에는 가중치가 3이다.

## - 관광지 주변 누비자 경로현황



[그림 21 - 관광지 주변 누비자 경로 1]

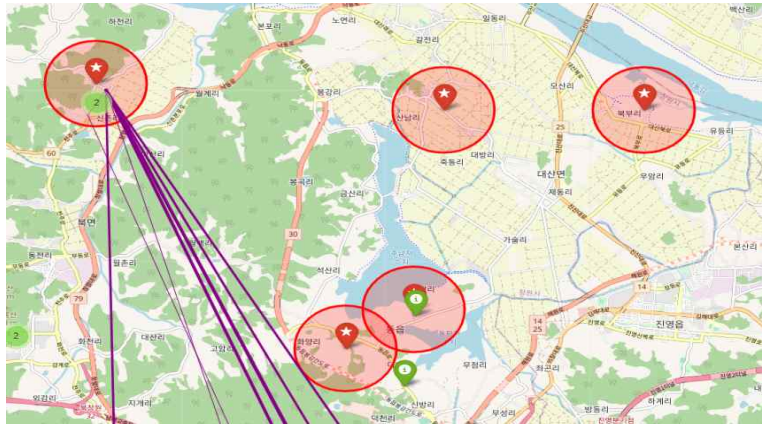


[그림 22 - 관광지 주변 누비자 경로 2]

### 1) 군집1

클러스터링의 결과로 “주남저수지”, “창원컨트리클럽”, “창원단감테마공원”, “주남저수지”, “마금산온천”, “북면수변생태공원” 군집1로 군집화되었다. 특이사항으로 터미널 168번 주변 관광지(마금산온천, 북면수변생태공원)가 존재한다.

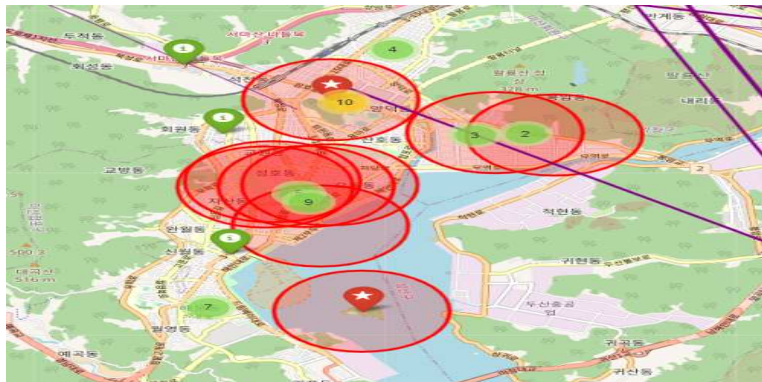




[그림 23 - 군집1]

## 2) 군집2

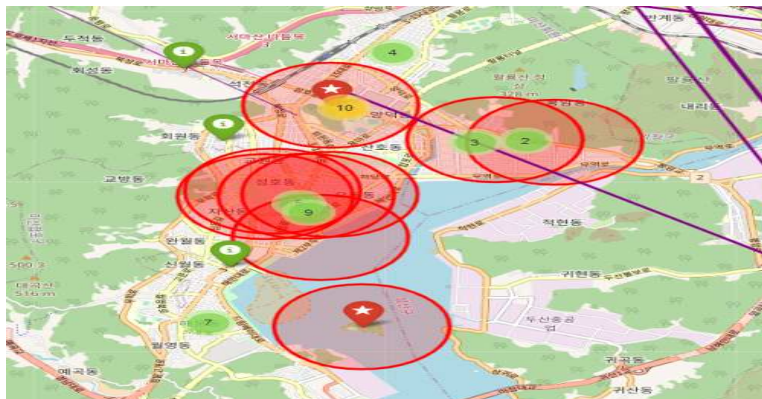
클러스터링의 결과로 “진해내수면환경생태공원”, “주기철목사기념관”, “진해해양공원”, “창원짚트랙”, “진해해양레포츠센터”, “제항산공원-모노레일”, “진해내수면환경생태공원”이 군집2로 군집화 되었다. 특이사항으로 터미널 222번 주변 관광지(“진해내수면환경생태공원”, “벚꽃길”)가 존재한다.



[그림 24 - 군집2]

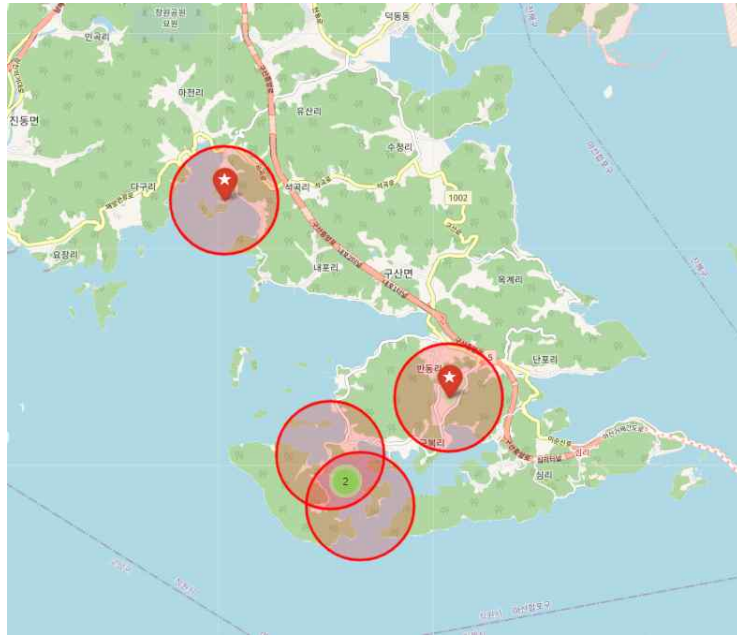
## 3) 군집3

클러스터링의 결과로 “봉암유원지”, “3.15아트”, “마산해양레포츠센터”, “돌섬유원지가” 군집3으로 군집화 되었다.



[그림 25 - 군집3]

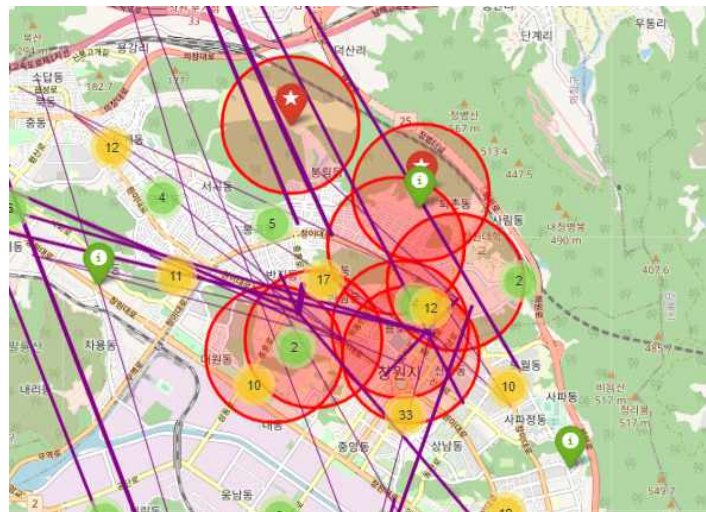
#### 4) 군집4



[그림 26 - 군집4]

클러스터링의 결과로 “콰이강의 다리 스카이워크”, “해양드라마세트장”, “로봇랜드”, “저도비치 로드”가 군집4로 군집화 되었다. 특이사항으로 군집화가 이루어진 관광지의 관광객이 창원시내 관광객의 다수임에도 불구하고 누비자 터미널이 존재하지 않는다.

#### 5) 군집5



[그림 27 - 군집5]

클러스터링의 결과로 창원컨트리클럽, 창원경륜장, 창원과학체험관, 성산아트홀, 용지호수공원 무빙보트, 창원의집, 독립미술관이 군집5로 군집화 되었다.



### 3. 활용방안 및 기대효과

#### - 시사점과 결과 값을 통한 결론

창원시의 자전거 교통사고 소식은 최근에도 보도되는 만큼 심심치 않게 들리고 있다. 시민들은 안전성 확보를 위해 자전거 도로 및 도로 개선이 필요하다고 목소리를 내고 있다. 그러나 창원시의 관계자들에 따르면 이를 어디서부터, 어떤 부분을, 어떻게 개선해야 할지 가늠하기 어려운 것이 실상이다. 거기에다가, 일부 관계자는 도로 개선이 필요하냐는 의문을 가지고 있다. 본 팀의 자료를 통해 누비자 이용경로에서 주요하거나 개선 시 효과를 볼 수 있는 곳을 빠르게 파악하는 등 관리감독의 효율성을 확보 할 수 있다. 나아가 추가적 도시 인프라 구축을 논의 할 수 있다.

전국 공용자전거를 조사하며 누비자에 대한 이용 현황과 예산, 전주시의 공용자전거 개선으로 인한 효과 그리고 전국 자전거 여행 지도를 확인할 수 있었다. 창원시는 최근 각종 드라마 촬영이 이루어졌고 문화관광 도시로서 많은 노력을 하고 있다. 하지만 전국 자전거 여행 도로에 창원시의 행정구 중 어느 하나 포함되어 있지 않은 안타까운 실정이다. 본 팀의 자료를 활용하여 창원시 내의 관광지 및 문화 공간을 홍보하고 누비자를 통한 여행 경로를 추천한다면 누비자의 이용률이 증가하고 창원시 지역경제 활성화와 관광객 유치와 같은 효과를 볼 수 있을 것이다. 나아가 누비자를 통한 관광에 필요한 예산을 논의해볼 수 있다.

#### - 터미널 이용객/경로

##### 1) 누비자 이동경로 데이터 활용방안

- ㄱ. 출퇴근 시간대에 왕래가 잦은 자전거 경로 중, 공단지부 도로를 개선 여부 판단이 가능하다. 최근 [참고자료 4.A](#)를 보면 공단 내부에서 자전거 사고가 빈번히 발생하기에 도로 개선이 필요하다.
- ㄴ. 공단지역의 도로 개선이 어렵다면 차선택으로 본 팀의 데이터를 활용하여 자전거 이용객에게 공단지부를 최소한으로 거쳐 가게 하거나, 공단지역의 안전한 도로로 우회하는 방안이 제시 가능하다.
- ㄷ. 의창구, 성산구는 터미널 밀집도가 높고 이용객 수가 많음에도 자전거 거리가 미흡한 곳을 확인할 수 있다. ([참고자료 4.B](#)) 따라서 창원 시내의 자전거 도로 관리 및 감독에서 본 팀의 데이터를 활용 가능하다.
- ㄹ. 지역별 이동경로와 이동량을 파악하여 거리가 긴 경로 중 도로가 복잡한 곳을 파악하는데 용이하다. 나아가 거리 조성 등의 인프라 구축을 고려해볼 수 있다.

##### 2) 누비자 이동경로 데이터 기대효과

- ㄱ. 기존 자전거 도로에 대한 안전성을 다시 한번 검토할 수 있고, 이를 통해 도로의 안전성을 높일 수 있다.
- ㄴ. 기존의 복잡하고 위험한 도로가 아닌 새롭고 안전한 자전거 경로 우회하여 자전거 이용객 뿐만 아니라, 해당 도로를 지나는 차량 운전자의 원활한 주행이 가능할 것으로 전망된다.
- ㄷ. 사람들이 많이 왕래하는 경로를 우선하여 도로를 관리할 수 있으므로 도로 개선 방향의 우선순위를 둘 수 있다. 이는 예산 할당에 도움이 될 것이라 예상된다.
- ㄹ. 왕래 횟수 대비 자전거 경로가 매우 긴 곳을 파악한다면, 그곳을 우선해서 관리하여 자전거 이용객의 이동 시간을 줄일 수 있다.

## - 관광지 데이터

### 1) 관광지 주변 데이터별 누비자 이용객 데이터 활용방안

- ㄱ. 사람들이 방문이 잦은 관광지의 자전거 도로 관리에 용이하다.
- ㄴ. 관광지 주변 터미널별 누비자 이용객 데이터를 이용하여 이용객이 많은 곳부터 우선해서 인프라를 개선하는데 활용 가능하다.
- ㄷ. 본 팀의 클러스터링 데이터를 토대로 한 추천 경로를 이용하여 자전거 이용에 미흡한 부분 개선에 활용할 수 있다.

※ 클러스터링 데이터를 활용한 본 팀의 추천 경로는 다음과 같음.

<1>

“주남저수지” → “창원단감테마공원” → “메타세콰이어 가로수길” → “동부마을 팽나무(우영우)” → “마금산온천”

<2>

“진해해양공원” → “창원짚트랙” → “주기철목사기념관” → “진해드림파크” → “진해해양레포츠센터” → “경화시장” → “제항산공원-모노레일” → “여좌천로망스다리” → “진해내수면환경생태공원” → “진해드림로드” → “창원편백치유의 숲”

<3>

“해양드라마세트장” → “로봇랜드” → “콰이강의다리 스카이워크” → “저도비치로드”

<4>

“창원경륜장” → “창원과학체험관” → “용지호수공원 무빙보트” → “성산아트홀” → “도립미술관” → “창원의 집” → “창원국제사격장” → “창원컨트리클럽”

<5>

“봉암유원지” → “굿데이뮤지엄” → “3.15 아트” → “가고파 꼬부랑길 벽화마을” → “문신미술관” → “창동예술촌” → “마산 오동동 통솔골목” → “마산해양레포츠 센터” → “돌섬유원지”

- ㄷ. 창원 시내 지정 관광지 외의 볼거리를 자전거길 추천 코스에 추가하여 본 팀이 추천한 경로 내부에 관광지로 지정되진 않았지만, 볼거리가 있다면 코스에 추가 가능하다. 또한 관광지로 가는 도중 감성 카페, 전통시장, 자연경관, 가로수길 등이 있다면 해당 코스에 추가 가능하다.

## - 관광지 주변 데이터별 누비자 이용객 데이터 기대효과

- ㄱ. 사람들이 많이 방문하는 관광지 코스로의 자전거 도로를 정비하면 해당 관광지에 찾아오는 관광객 수가 증가할 것으로 전망된다.(참고자료 4.D.) 나아가 창원 시민뿐 아닌, 타 지역 시민, 외국인 등 외부인의 자전거 방문객의 증가를 기대한다. 이로 인해 누비자 수익이 증가할 것을 기대해 볼 수 있다.
- ㄴ. 자전거 관광지 코스 내의 지역 활성화를 유도 할 것으로 기대한다. 만약 자전거 코스를 통해 유입된 관광객이 늘어난다면, 기존 창원시가 지정한 관광지뿐만 아니라 다른 관광지가 추가로 생길 가능성을 기대할 수 있다. 기존 창원시 내의 추천 자전거 경로에서 추가적으로 다양한 관점의 자전거 도로 추천할 수 있고, 이는 기존 관광지 자전거 코스 외에 다른 코스를 원하는 사람들에게 본 팀이 추천하는 코스를 제공함으로써 더 많은 창원 시내 관광지를 방문하게 할 수 있다.

### 3. 활용데이터 및 참고 문헌 출처 등

---

1. 창원시 빅데이터 (<https://bigdata.changwon.go.kr/portal/dataset/datasetList.do>)
  - A. 창원시 무인대여 공영자전거 누비자 대여\_반납 이력(2018년)
  - B. 경상남도 창원시\_누비자 이용현황
  - C. 경상남도 창원시\_누비자 터미널
  - D. 경상남도 창원시\_주요관광지점 입장객
  - E. 주요 관광지 통계(2021년)
  - F. 사망교통사고 정보
    - i. 자전거 교통사고 정보 발췌
  - G. 경상남도 창원시\_관광현황
2. 누비자 홈페이지 내부 터미널 (<https://www.nubija.com/terminal/terminalList.do>)
  - A. nubibike
    - i. 터미널 목록을 크롤링을 통하여 데이터 생성
3. 창원시경륜공단\_무인대여 공영자전거 누비자 현황(<https://www.data.go.kr/data/15014737/fileData.do>)
  - A. 주기성 과거데이터 목록에서 2018년도 전체 데이터 수집
4. 그 외 참고 자료
  - A. KBS 뉴스(2022.09.20): 창원서 또 우회전 사망 사고… 3년전 같은 장소  
<https://www.youtube.com/watch?v=87cDnZ-Zcp4>
  - B. 경남도민일보(2022.03.11): 경남 창원 끊어진 자전거길, 그리고 사망사고  
<https://www.youtube.com/watch?v=dKV3Y49Z9iI>
  - C. 아름다운 자전거길 100선  
<https://www.bike.go.kr/content.do?key=2008069223071>
  - D. 전주시 여행은 자전거로 공영자전거 이용객 급증  
<https://www.yna.co.kr/view/AKR20220802077000055>
  - E. 창원관광 홈페이지  
<https://culture.changwon.go.kr/index.changwon?contentId=9>