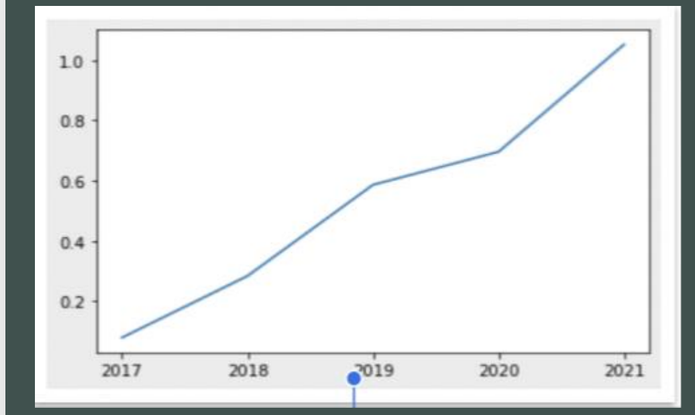
전기차 충전소 최적 입지 선정

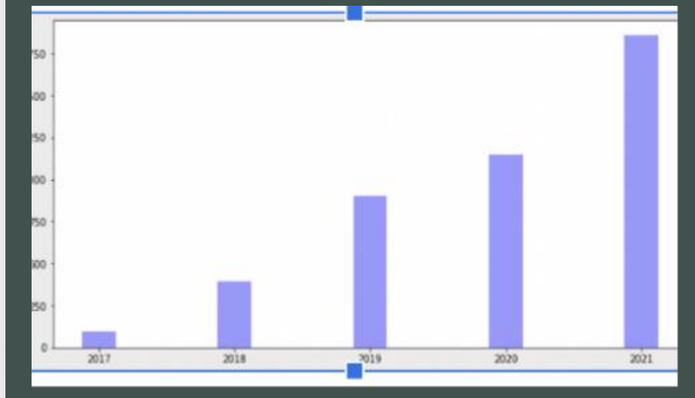
데이터 청년 캠퍼스 홍익대 1팀

성정현 성진현 최주영
이세린 최민지

01 프로젝트 기획 배경

01 증가하는 전기차 수요





02 수요대비 부족한 충전 시설

시급한 개선이 필요한 전기차 충전 정책(단위: %)

자료:소프트베리

전기차 충전시설 보급 확대 40

미흡한 전기차 충전시설 관리 32

급속충전기 보급 확대 21

전기차 충전시설 정보의 실시간 확인 7

대상 : 소프트베리 앱 사용자 1896명

03 전기차 충전시설 확대 정책 추진

정부세종청사, 전기차 충전 편해진다

- 2024년까지 총 110기 구축 예정 -

□ 행정안전부 정부청사관리본부(본부장 조소연)는 **전기자동차 보급 확대***에 따라 2024년까지 정부세종청사에 **전기차 충전기 110기**를 **확충**할 계획이라고 밝혔다.

* 국내 전기차 누적 보급량 : '18년 5.6만 → '20년 13.5만 → '25년 113만 → '30년 300만 (자료출처 : 산업통상자원부, 제4차 친환경자동차 기본계획)

○ 정부는 **기후변화 대응 및 자동차 온실가스 배출 감축***을 위해 전기차 보급을 확대해 가면서 친환경 자동차법에 따라 **전기차 충전기 설치를 지자체 조례****에 정하여 의무적으로 설치**토록 하고 있다.

- 정부세종청사가 위치한 세종시는 법정 주차면의 0.5% 이상을 설치토록 규정하고 있고, 향후 2%까지 확대할 예정이다.

* 온실가스 배출 감축('17년 대비, 주행기준) : '25년까지 8% → '30년까지 24%

** 세종특별자치시 환경친화적 자동차의 보급촉진 및 이용활성화에 관한 조례 제10조

목차
Contents

01 프로젝트 개발 환경


colab


python™


Se
Selenium


Folium


OPTUNA


pandas


B
Bootstrap


Windows 10

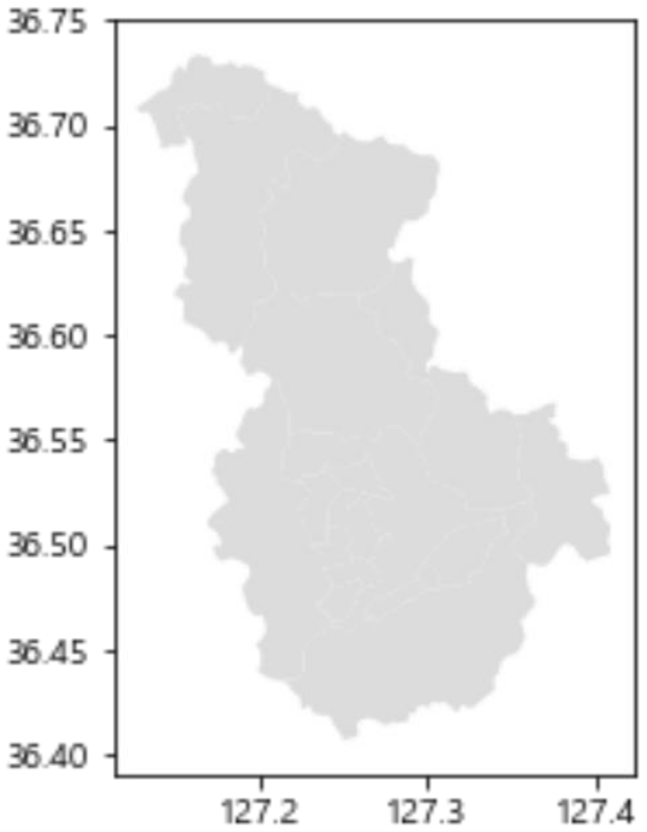

pydeck

01 프로젝트 개발 흐름도

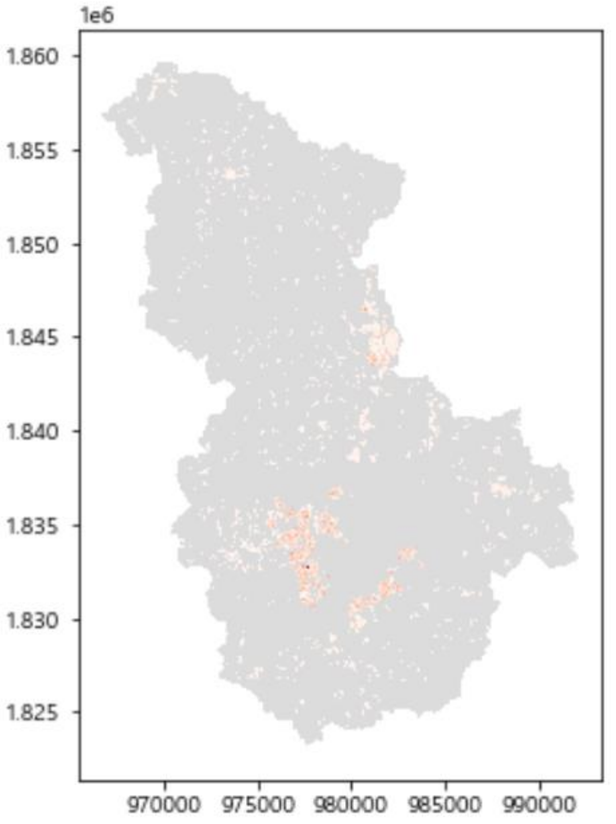
Step 01.	Step 02.	Step 03.	Step 04.
데이터 수집 및 전처리	**데이터 시각화 및 탐색**	**데이터 분석**	**웹**
전국 충전소 리스트	Folium 지도 시각화	Random Forest	bootstrap LiveServer Port Forwarding
전국주차장정보표준데이터	Matplotlib 지도, 데이터 시각화	LightGBM	kakao Maps API 외부 접속 구현
국토정보플랫폼 국토정보맵	Seaborn 데이터 시각화	CATBoost	
전국 건축물대장	Pydeck 3D 시각화		
전국 행정동 경계			
고저차 수직기준변환 KNGeoid18			

데이터 전처리

WGS84 좌표계를 사용했을 때



GRS80 직각좌표계를 사용했을 때



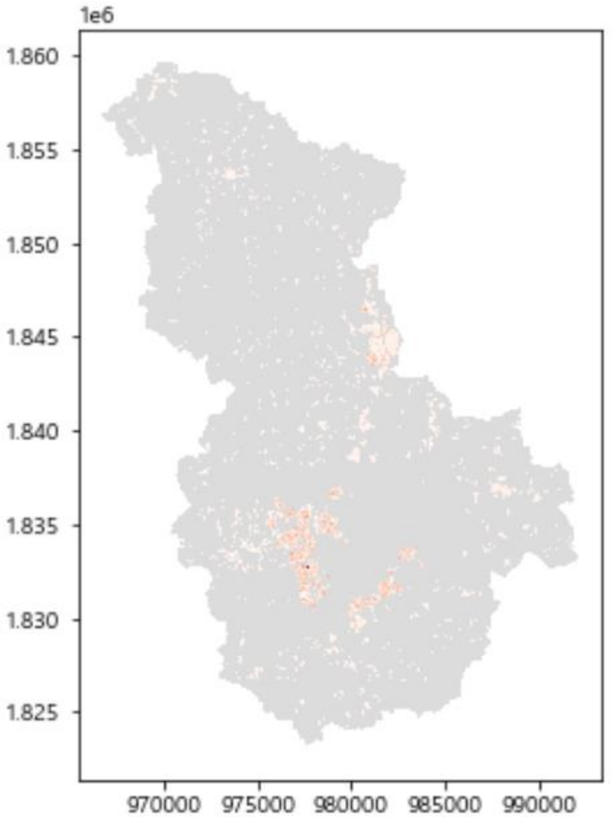
데이터 전처리

인구수 격자 100m x 100m
건축물수 격자 100m x 100m
주거용도면적 격자 100m x 100m
건축물 높이 격자 100m x 100m
건축물 연면적 격자 100m x100m

다음 격자 데이터들은
GRS80 좌표계를 사용하고 있으므로
WGS84 좌표계와 통일이 필요

* 국토정보플랫폼 국토정보맵

GRS80 직각좌표계를 사용했을 때



데이터 전처리

```
# 가장 왼쪽의 폴리곤 시작점
poly_left = 2000000
for i, g in enumerate(df['geometry']):
    if int(g.split()[1][2:]) < poly_left:
        poly_left = int(g.split()[1][2:])

# 가장 아래의 폴리곤 시작점
poly_down = 3000000
for i, g in enumerate(df['geometry']):
    if int(g.split()[2][:-1]) < poly_down:
        poly_down = int(g.split()[2][:-1])

# 가장 위의 폴리곤 시작점
poly_up = 0
for i, g in enumerate(df['geometry']):
    if int(g.split()[4][:-1]) > poly_up:
        poly_up = int(g.split()[4][:-1])

# 가장 오른쪽의 폴리곤 시작점
poly_right = 0
for i, g in enumerate(df['geometry']):
    if int(g.split()[5]) > poly_right:
        poly_right = int(g.split()[5])

print('poly_left', poly_left)
print('poly_down', poly_down)
print('poly_up', poly_up)
print('poly_right', poly_right)
```

```
poly_left 966700
poly_down 1823200
poly_up 1859600
poly_right 992100
```

```
# 세종시는 결과의 위도와 경도 범위에 있음

latitude = []
longitude = []

for g in geometry:
    for l in g[16:-3].split(', '):
        longitude.append(l.split()[0])
        latitude.append(l.split()[1])

print('latitude : ', min(latitude), "~", max(latitude))
print('longitude : ', min(longitude), "~", max(longitude))
```

```
latitude :  36.40675835257603 ~ 36.73376199366808
longitude :  127.12765872447403 ~ 127.4092754044858
```

세종시는 254 * 364의 격자로 이루어져 있음
WGS84 좌표계와 GRS80 좌표계의
끝 점들을 각각 매칭시켜
GRS80 좌표계로 통일

데이터 전처리

```
GCS = []

def get_location(address):
  url = 'https://dapi.kakao.com/v2/local/search/address.json?query=' + address
  headers = {"Authorization": "KakaoAK 3c768412ed253db7abaf7e0d5a15788f"}
  api_json = json.loads(str(requests.get(url,headers=headers).text))
  address = api_json['documents'][0]['address']
  crd = {"lat": str(address['y']), "lng": str(address['x'])}
  address_name = address['address_name']

  return crd

for i, addr in enumerate(ev_charger['주소']):
    addr = ev_charger['주소'].loc[i].split(',')[0].split('(')[0]
    while(1):
        try:
            crd = get_location(addr)
            #print(i, crd['lat'], crd['lng'])
            break
        except:
            addr = addr[:-1]

    GCS.append((crd['lat'], crd['lng']))
```

kakao developers

위경도(WGS84 좌표계)를 GRS80 좌표계로 변환

```
def WGS84_to_GRS80(WGS):
    GRS = []

    # 한 격자당 위도 변화
    length_100m = (36.73376199366808 - 36.40675835257603) / 364

    # 한 격자당 경도 변화
    width_100m = (127.4092754044858 - 127.12765872447403) / 254

    for i, w in enumerate(WGS):
        GRS_x = 966700 + (float(w[1]) - 127.12765872447403)/width_100m * 100
        GRS_y = 1823200 + (float(w[0]) - 36.40675835257603)/length_100m * 100
        GRS.append((GRS_x, GRS_y))

    return GRS
```

세종시는 254 * 364의 격자로 이루어져 있음
WGS84 좌표계와 GRS80 좌표계의
끝 점들을 각각 매칭시켜
GRS80 좌표계로 통일

데이터 전처리

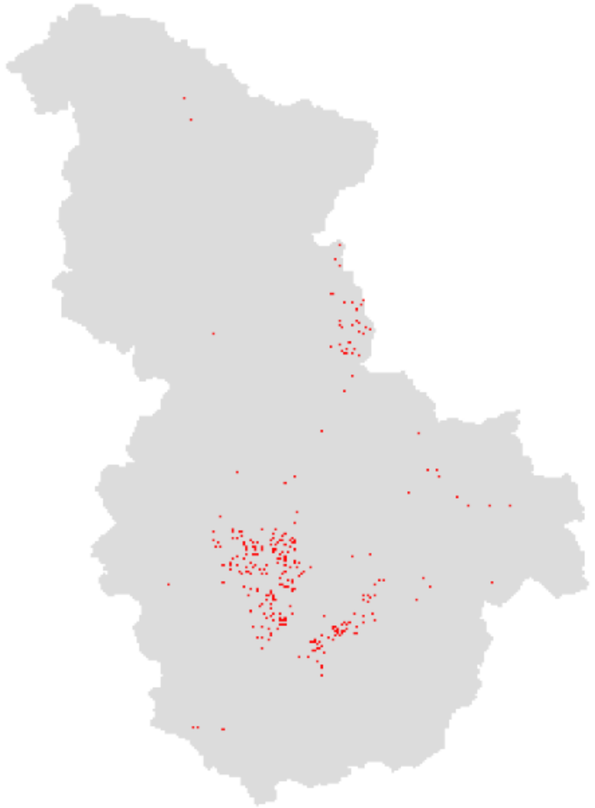
운영기관	충전소	충전기 ID	충전기 타입	지역	시군구	주소	이용가능시간	이용자 제한	충전용량
환경부(한국자동차환경협회)	신북보건지소공영주차장	21	DC콤보	전라남도	영암군	전라남도 영암군 신북면 간은정로 21-7	24시간 이용가능	NaN	급속(200kW동시)
환경부(한국자동차환경협회)	신북보건지소공영주차장	22	DC콤보	전라남도	영암군	전라남도 영암군 신북면 간은정로 21-7	24시간 이용가능	NaN	급속(200kW동시)
환경부(한국자동차환경협회)	한국생산기술연구원 대경본부바이오메디칼생산기술센터	1	DC콤보	경상북도	영천시	경상북도 영천시 양호길 59	24시간 이용가능	NaN	급속(100kW단독)

운영기관	충전소	충전기 ID	충전기타입	지역	시군구	주소	이용가능시간	이용자 제한	충전용량	비고	WGS84	GRS80
환경부(한국자동차환경협회)	LH 세종본부	11	DC콤보	세종특별자치시	가름로	세종특별자치시 가름로 238-1	24시간 이용가능	NaN	급속(200kW동시)	NaN	(36.4951836975395, 127.265999510588)	(979177.4426257282, 1833042.9563228148)
환경부(한국자동차환경협회)	고복저수지 연기대첩비 공원 주차장	1	DC콤보	세종특별자치시	연서면	세종특별자치시 연서면 용암리 208-5	24시간 이용가능	NaN	급속(100kW멀티)	NaN	(36.5995423567613, 127.227895171906)	(975740.6781468542, 1844659.5095299517)

데이터 전처리

격자 데이터에 충전소가 존재하는지 여부를 매칭

```
EV = []

for g in material['geometry']:
    g = g.split()
    # print(g[1][2:], g[2][:-1], g[4][:-1], g[5]) # 폴리곤의 좌측x, 하단y, 상단y, 우측x 좌표
    temp = 0
    for grs in ev_charger['GRS80']:
        x = int(float(grs[0]))
        y = int(float(grs[1]))

        if x>int(g[1][2:]) and x<int(g[5]) and y>int(g[2][:-1]) and y<int(g[4][:-1]):
            temp = 1
            break

    EV.append(temp)
```



충전소가 격자 내에 있는지 없는지 탐색하여
격자 내 충전소 여부 확인

데이터 전처리

	주용도코드명	대지위치
0	교육연구시설	세종특별자치시 반곡동 42-9번지
1	주거지	세종특별자치시 반곡동 146-75번지
2	농축산업시설	세종특별자치시 반곡동 146-79번지
3	농축산업시설	세종특별자치시 반곡동 146-79번지
4	창고시설	세종특별자치시 반곡동 146-79번지

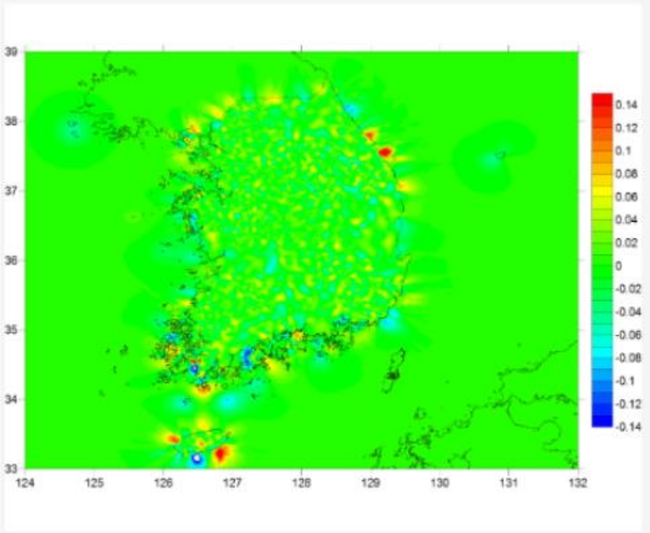
	주용도코드명	대지위치	WGS84	GRS80
0	교육연구시설	세종특별자치시 반곡동 42-9번지	(36.4957725586472, 127.312070498707)	(983332.7472695216, 1833108.5046581437)
1	주거지	세종특별자치시 반곡동 146-75번지	(36.4872193967093, 127.317747622221)	(983844.7870295589, 1832156.4201691155)
2	농축산업시설	세종특별자치시 반곡동 146-79번지	(36.4882209142039, 127.318265076261)	(983891.4580314871, 1832267.9028323719)
3	농축산업시설	세종특별자치시 반곡동 146-79번지	(36.4882209142039, 127.318265076261)	(983891.4580314871, 1832267.9028323719)
4	창고시설	세종특별자치시 반곡동 146-79번지	(36.4882209142039, 127.318265076261)	(983891.4580314871, 1832267.9028323719)

주차장명	주차장구분	소재지도로명주소
아름주차장	민영	세종특별자치시 보듬3로 104-7
원프라자	민영	세종특별자치시 보듬3로 8-37
도담프라자	민영	세종특별자치시 보듬3로 8-5
온누리타워	민영	세종특별자치시 새롬중앙로 63
무지개타워	민영	세종특별자치시 새롬중앙로 41

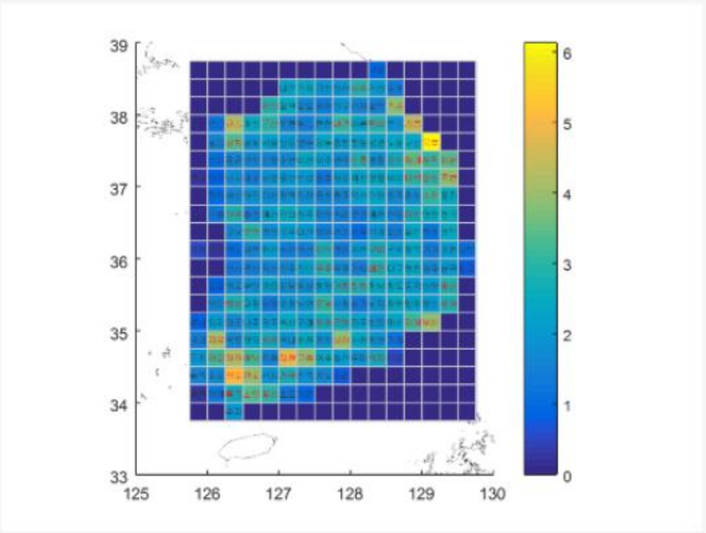
주차장명	주차장구분	소재지도로명주소	소재지지번주소	주차구획수	위도	경도	GRS80
아름주차장	민영	세종특별자치시 보듬3로 104-7	세종특별자치시 아름동 1287번지	54	36.512321	127.248429	(977592.6964064469, 1834950.5736431568)
원프라자	민영	세종특별자치시 보듬3로 8-37	세종특별자치시 도담동 678번지	107	36.515574	127.257844	(978441.8684085815, 1835312.6772564517)
도담프라자	민영	세종특별자치시 보듬3로 8-5	세종특별자치시 도담동 668번지	127	36.512822	127.258317	(978484.5299441109, 1835006.3418295265)
온누리타워	민영	세종특별자치시 새롬중앙로 63	세종특별자치시 새롬동 산25번지	50	36.486537	127.250254	(977757.2635052346, 1832080.431094084)
무지개타워	민영	세종특별자치시 새롬중앙로 41	세종특별자치시 새롬동 341-95번지	231	36.484534	127.251234	(977845.6621753671, 1831857.510848436)

데이터 전처리

세종시 지형 고저차 데이터 생성 →



KNGeoid18 지오이드 오차



도엽별 지오이드 오차

세종시 격자별 차량수 데이터 생성 ↓

	구분	합계	1인당차량수
0	고운동	16236	2.341956
1	금남면	5289	1.334279
2	다정동	9855	3.142973
3	대평동	5522	2.064832
4	도담동	28642	1.110153

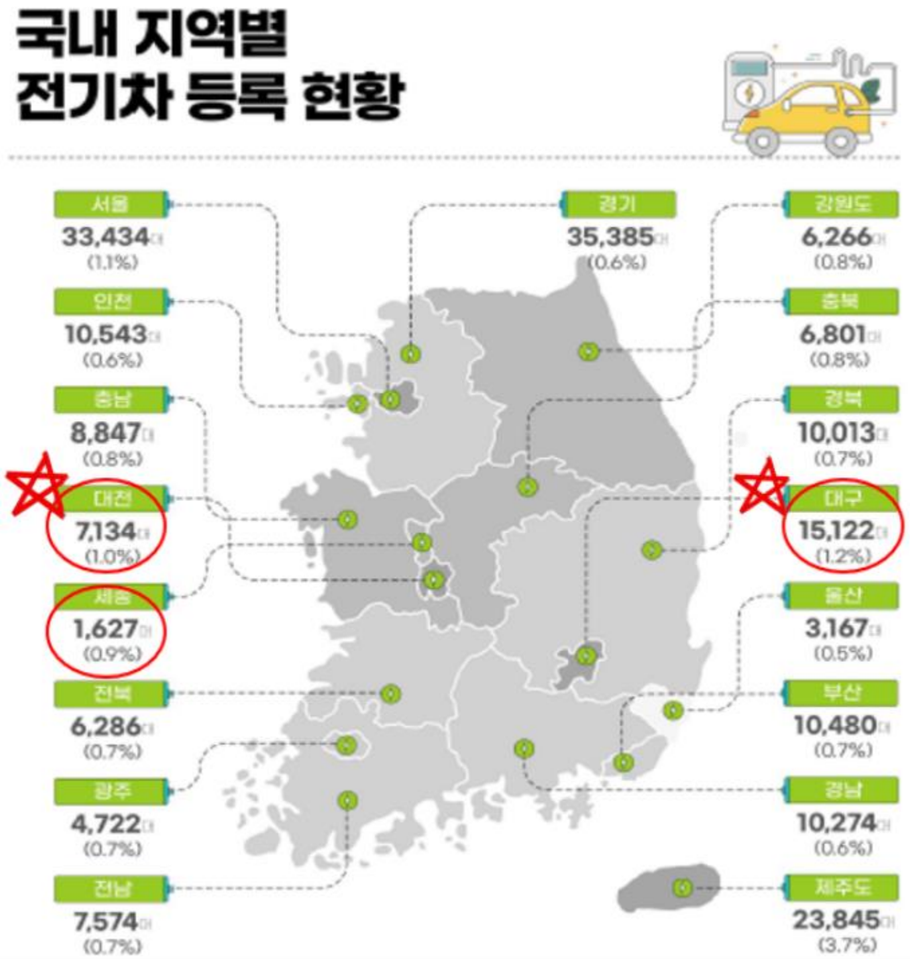
데이터 전처리

	geometry	lat	lng	읍면동	고저차	인구수	건축물수	차량수	주거용도면적	건축물높이	...	주차장수	교육연구시설	주거지	농축산업시설	창고시설	편의시설	생산시설	님비시설	자동차관련시설	EV
0	POLYGON ((977800.00000 1828600.00000, 977800.0...	36.455719	127.251282	금남면	24.869	0.0	1.0	0.000000	83.64	0.00	...	0	0	0	0	0	0	0	0	0	0
1	POLYGON ((990100.00000 1837300.00000, 990100.0...	36.533877	127.387655	부강면	25.167	0.0	1.0	0.000000	208.91	7.80	...	0	0	0	0	0	0	0	0	0	0
2	POLYGON ((980600.00000 1843000.00000, 980600.0...	36.585083	127.282326	연서면	24.825	27.0	4.0	38.518751	99.70	6.43	...	0	0	0	0	0	0	0	0	0	0
3	POLYGON ((983800.00000 1830600.00000, 983800.0...	36.473686	127.317805	금남면	25.007	0.0	3.0	0.000000	123.00	5.37	...	0	0	0	0	0	0	0	0	0	0
4	POLYGON ((969300.00000 1857700.00000, 969300.0...	36.717142	127.157040	소정면	24.447	27.0	20.0	28.287931	77.54	8.25	...	0	0	0	0	0	0	0	0	0	0

5 rows × 21 columns

세종시 전처리 결과 47396 행 x 21 열

데이터 전처리



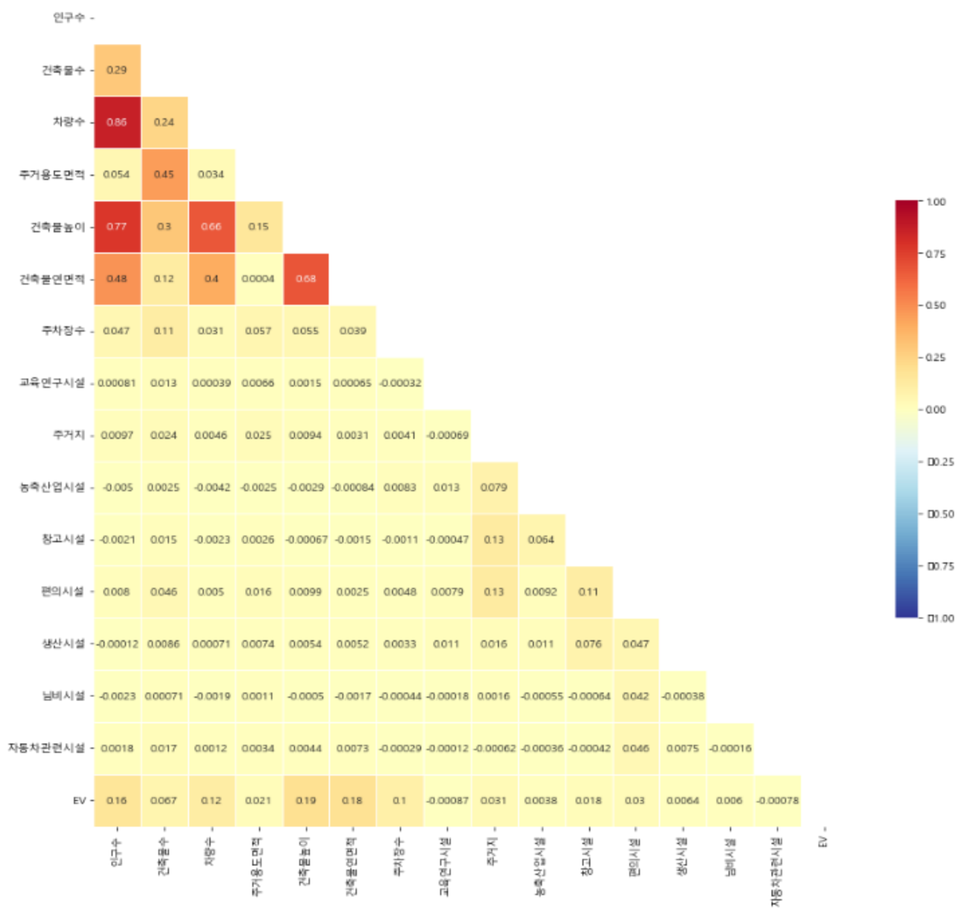
	geometry	lat	lng	읍면동	고저차	인구수	건축물수
0	POLYGON ((1098300.00000 1762500.00000, 1098300...	35.854989	128.587857	남구	28.803	124.0	18
1	POLYGON ((1095800.00000 1760400.00000, 1095800...	35.836205	128.560049	남구	28.785	298.0	47
2	POLYGON ((1099700.00000 1760500.00000, 1099700...	35.837099	128.603430	남구	28.888	173.0	6

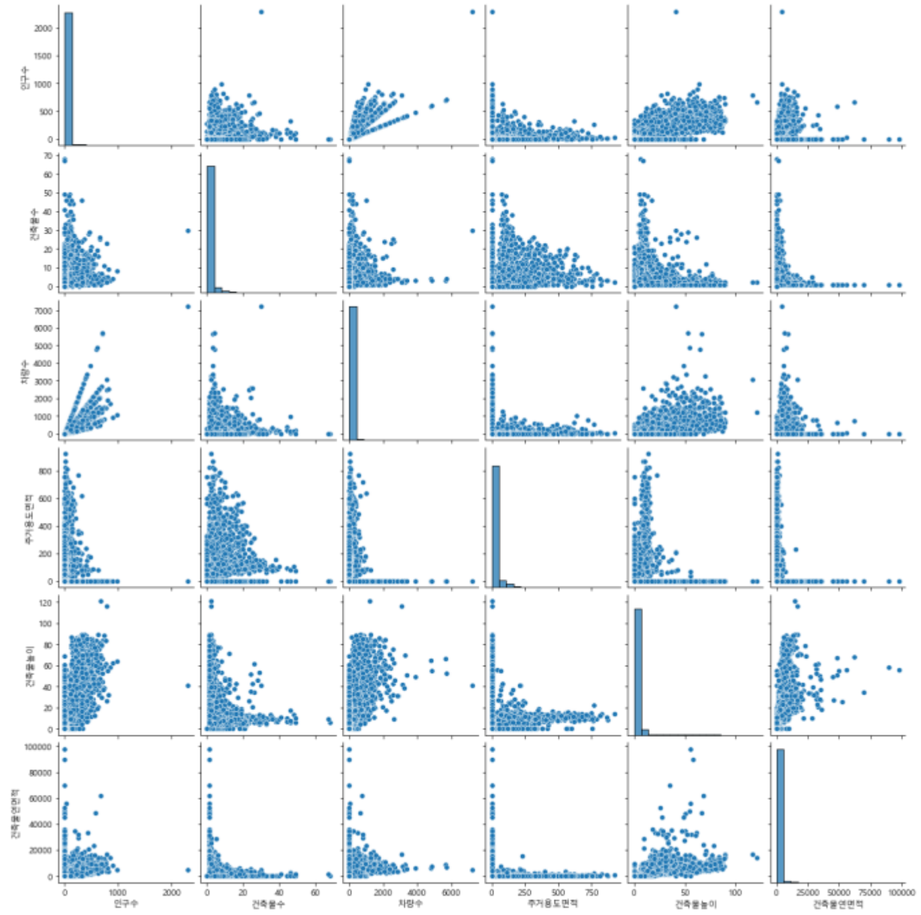
대구광역시 90806 x 21

	geometry	lat	lng	읍면동	고저차	인구수	건축물수
0	POLYGON ((992800.00000 1826000.00000, 992800.0...	36.431711	127.419101	대덕구	25.362	77.0	8
1	POLYGON ((993300.00000 1822700.00000, 993300.0...	36.402060	127.424605	대덕구	25.437	286.0	1
2	POLYGON ((992800.00000 1827500.00000, 992800.0...	36.445189	127.419101	대덕구	25.342	81.0	18

대전광역시 56081 x 21

모델링

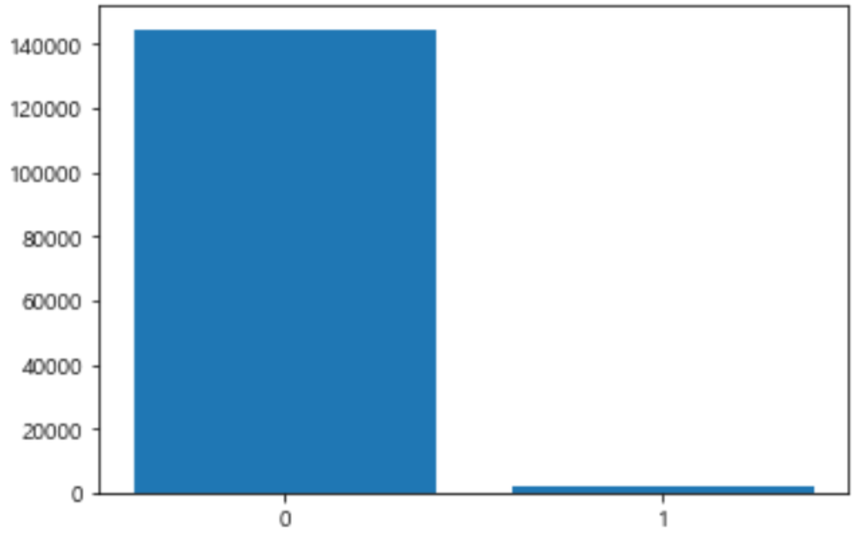

인구수
건축물수
차량수
주거용도면적
건축물높이
건축물연면적
주차장수
교육연구시설
주거지
농축산업시설
창고시설
편의시설
생산시설
남비시설
자동차관련시설
EV
0.29
0.86 0.24
0.054 0.45 0.034
0.77 0.3 0.66 0.15
0.48 0.12 0.4 0.0004 0.68
0.047 0.11 0.031 0.057 0.055 0.039
0.00081 0.013 0.00039 0.0066 0.0015 0.00065 -0.00032
0.0097 0.024 0.0046 0.025 0.0094 0.0031 0.0041 -0.00069
-0.005 0.0025 -0.0042 -0.0025 -0.0029 -0.00084 0.0083 0.013 0.079
-0.0021 0.015 -0.0023 0.0026 -0.00067 -0.0015 -0.0011 -0.00047 0.13 0.064
0.008 0.046 0.005 0.016 0.0099 0.0025 0.0048 0.0079 0.13 0.0092 0.11
-0.00012 0.0086 0.00071 0.0074 0.0054 0.0052 0.0033 0.011 0.016 0.011 0.076 0.047
-0.0023 0.00071 -0.0019 0.0011 -0.0005 -0.0017 -0.00044 -0.00018 0.0016 -0.00055 -0.00064 0.042 -0.00038
0.0018 0.017 0.0012 0.0034 0.0044 0.0073 -0.00029 -0.00012 -0.00062 -0.00036 -0.00042 0.046 0.0075 -0.00016
0.16 0.067 0.12 0.021 0.19 0.18 0.1 -0.00087 0.031 0.0038 0.018 0.03 0.0064 0.006 -0.00078
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


인구수
건축물수
차량수
주거용도면적
건축물높이
건축물연면적

모델링

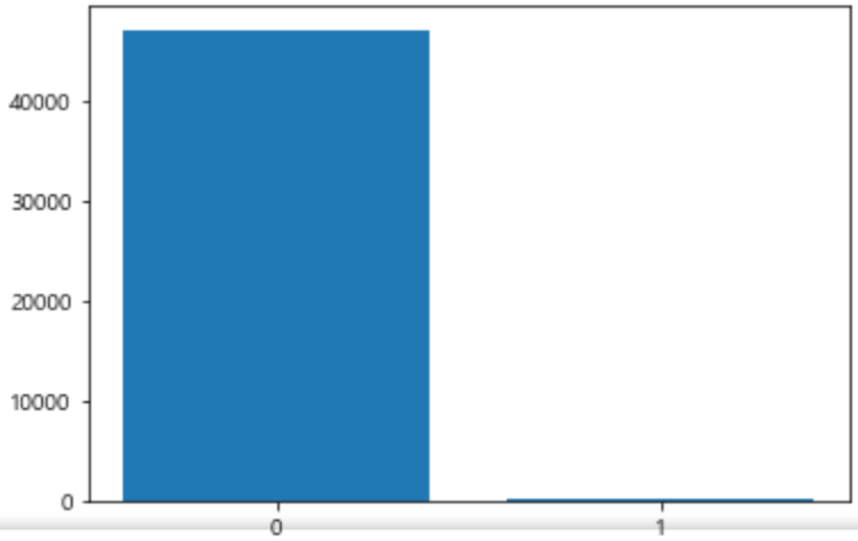
Train set

대전 + 대구 = 146,887 격자, 2,251개 충전소
0 : 1 = 0.985 : 0.015

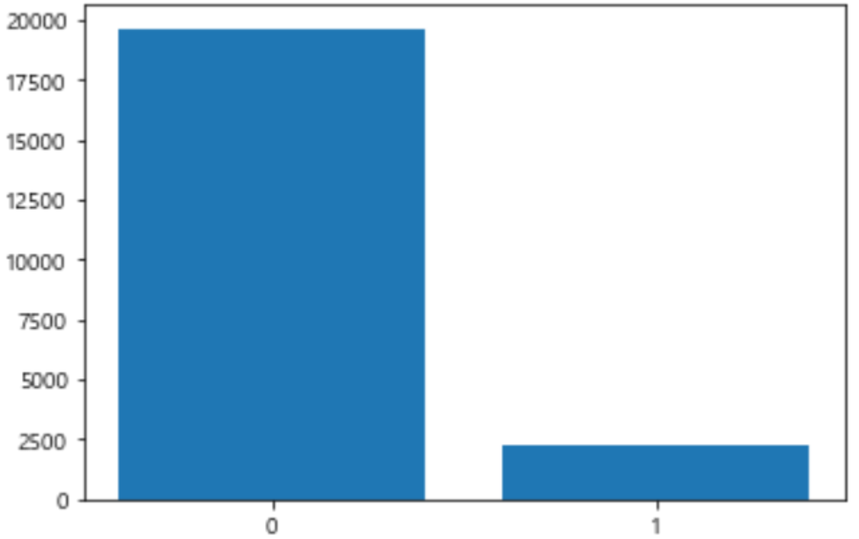


Test set

세종 = 47,396 격자, 268개 충전소
0 : 1 = 0.993 : 0.007



모델링

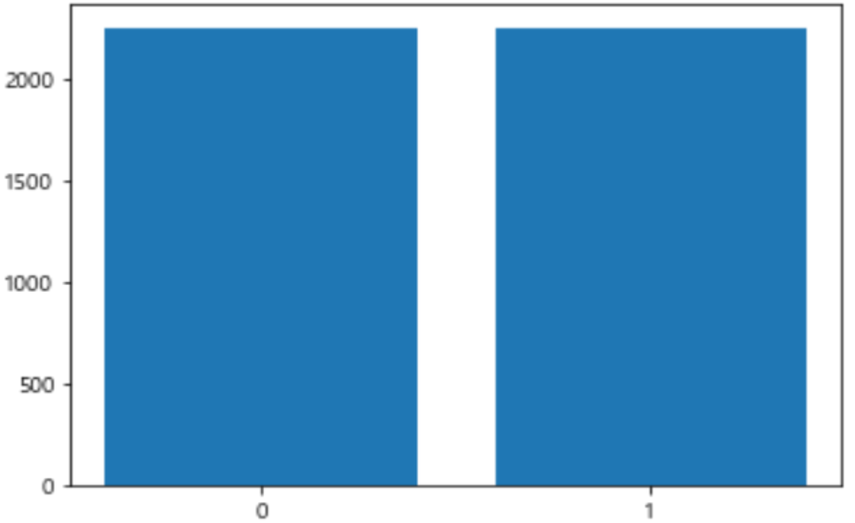


Train set

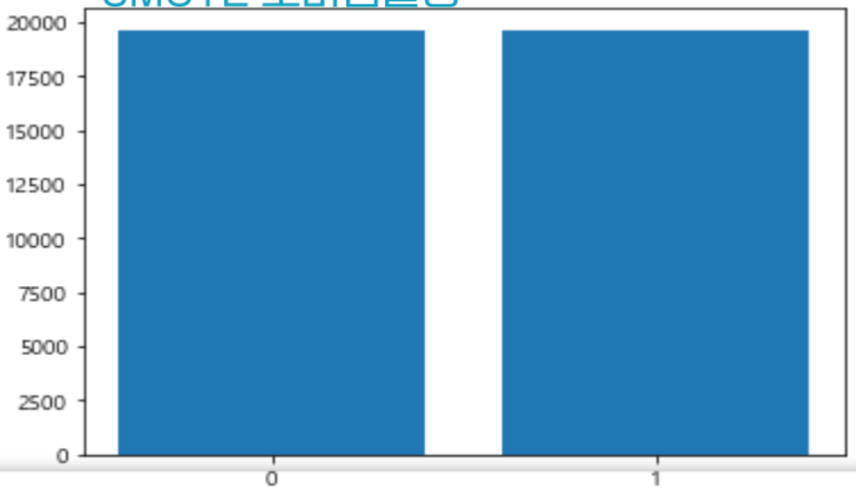
인구수 = 0, 충전소 = 0인 격자 제거

0 : 1 = 0.897 : 0.103

언더샘플링



SMOTE 오버샘플링



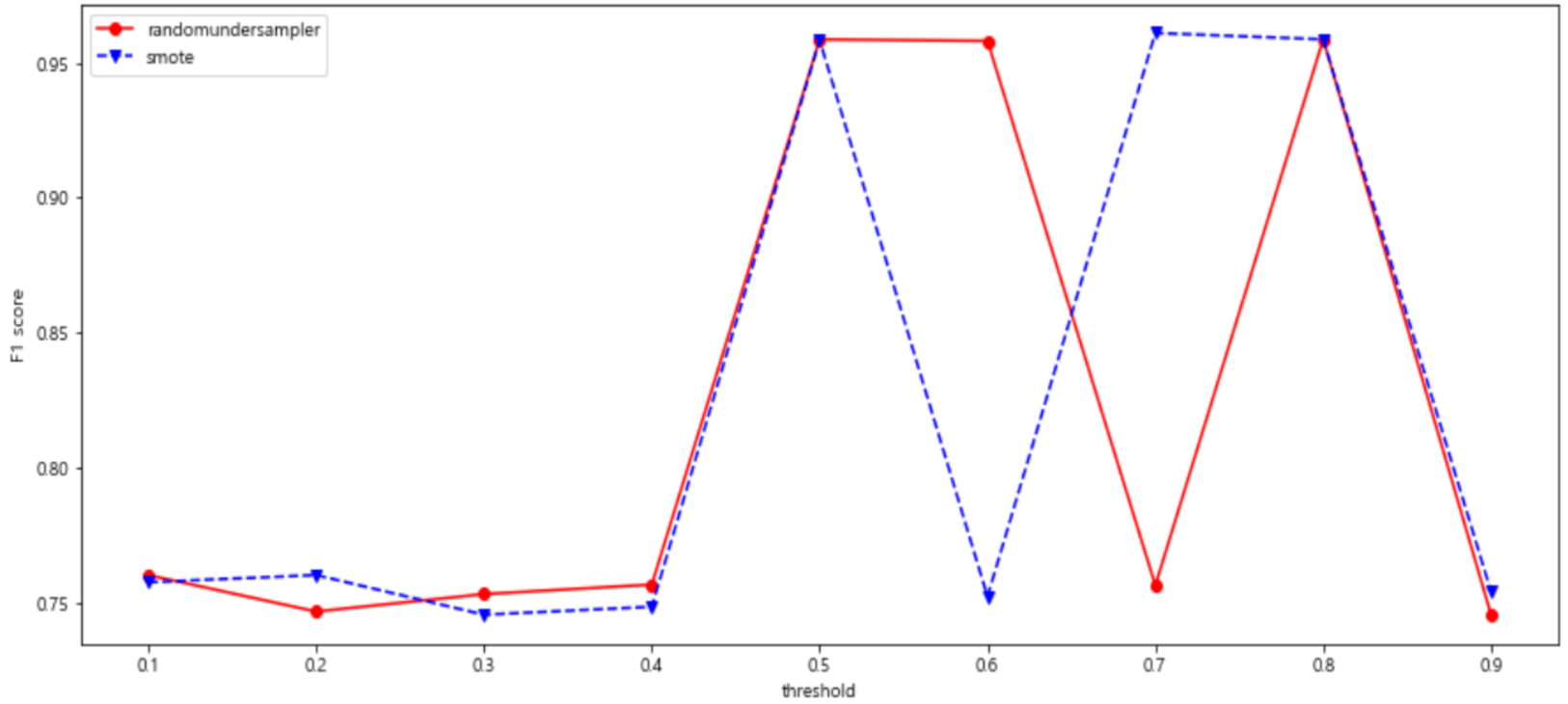
모델링

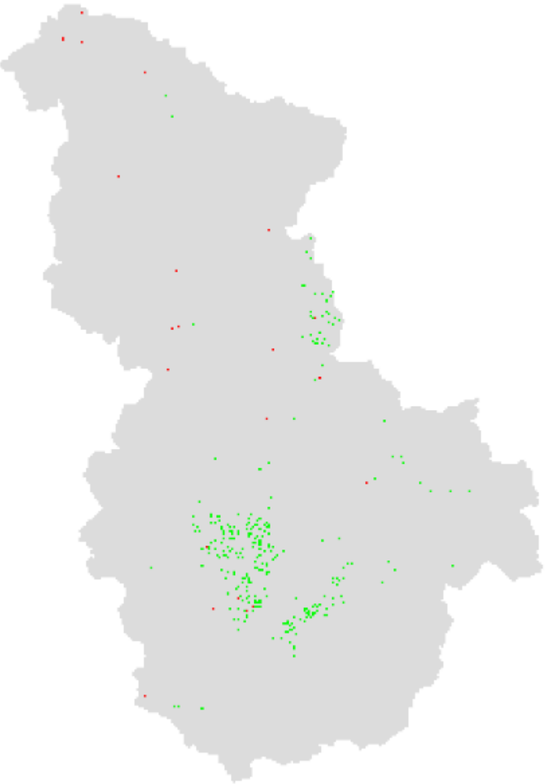
	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

충전소가 없는 격자에
충전소가 있어야 한다고 모델이 분류한 경우

False Positive = 최적 입지로 선정

모델링 - Random Forest





F1 score가 0.9 넘는 모델들의 추천 입지는 총 22곳

모델링

Precision

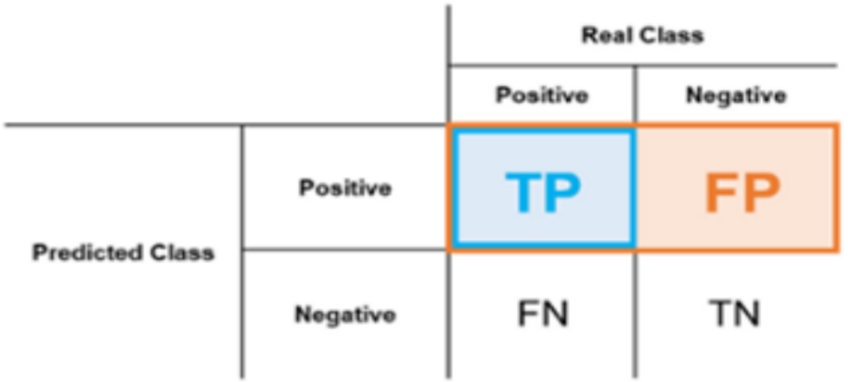
Of all positive predictions,
how many are really positive?

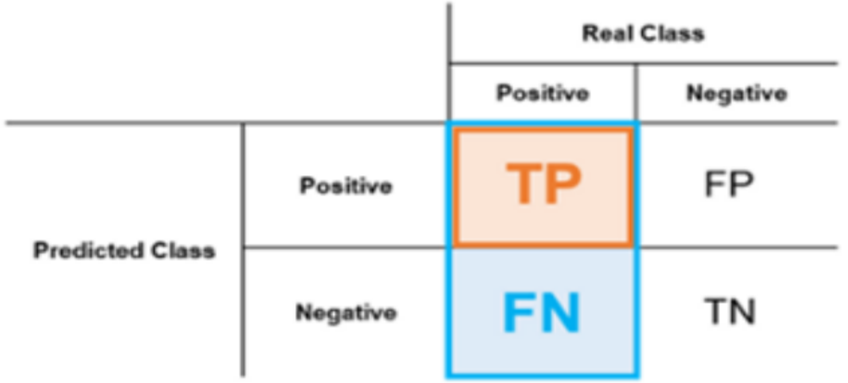
$$\frac{TP}{TP + FP}$$

Recall

Of all real positive cases,
how many are predicted positive?

$$\frac{TP}{TP + FN}$$

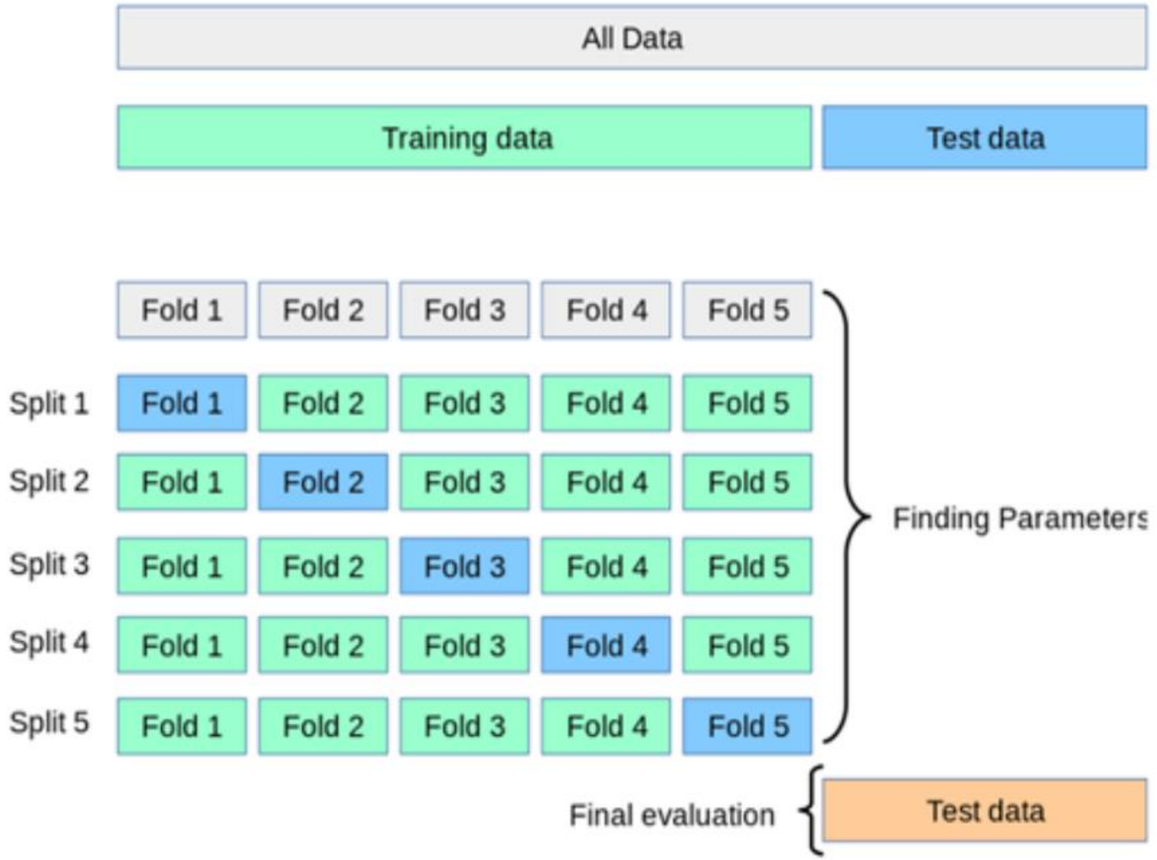




Zeya. 2021

데이터가 매우 불균형하므로 재현율과 정밀도를 활용한 F1 score와
Micro 평균을 활용한다.

모델링



GridSearchCV를 활용하려 했지만

K-Fold Cross Validation에서
불균형한 데이터셋은

데이터가 고르게 분포되지 않을 수
있는 문제가 있음

모델링

+

CV를 사용하지 않고
최적의 파라미터를 찾아주기 위해 AutoML 기법인 OPTUNA를 적용

모델링

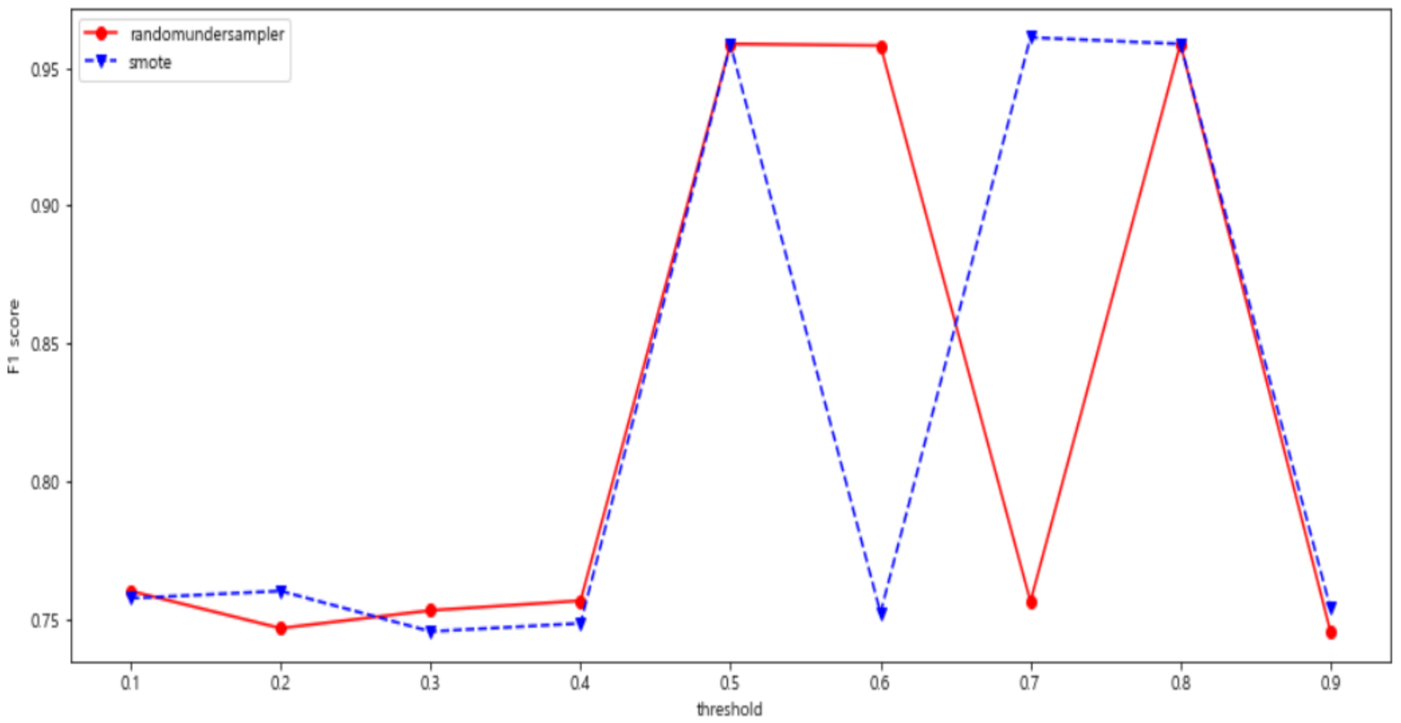
```python
def lgbm_objective(trial):

    params = {
            "booster": 'gbtree',
            "max_depth": trial.suggest_int("max_depth", 4, 16),
            "learning_rate": trial.suggest_uniform('learning_rate', 0.0001, 0.99),
            'n_estimators': trial.suggest_int("n_estimators", 1000, 10000, step=100),
            "colsample_bytree": trial.suggest_float("colsample_bytree", 0.5, 1.0),
            "colsample_bylevel": trial.suggest_float("colsample_bylevel", 0.5, 1.0),
            "colsample_bynode": trial.suggest_float("colsample_bynode", 0.5, 1.0),
            "reg_lambda": trial.suggest_loguniform("reg_lambda", 1e-2, 1),
            "reg_alpha": trial.suggest_loguniform("reg_alpha", 1e-2, 1),
            'min_child_weight': trial.suggest_int('min_child_weight', 2, 15),
            "gamma": trial.suggest_float("gamma", 0.1, 1.0, log=True),
        }

    clf = lightgbm.LGBMClassifier(**params)
    clf.fit(train_data, train_target, verbose=0)
    proba = clf.predict_proba(test_data)
    preds = (proba[:,1] >= threshold).astype('int')
    f1 = f1_score(test_target, preds)

    return f1

study = optuna.create_study(direction='maximize')
study.optimize(lgbm_objective, n_trials=50, n_jobs=4, show_progress_bar=True)

print("Number of finished trials: ", len(study.trials))
print("Best trial:")

trial = study.best_trial

print("  F1 : {}".format(trial.value))
print("  Best hyperparameters: ")

for key, value in trial.params.items():
    print("    {}: {}".format(key, value))
```

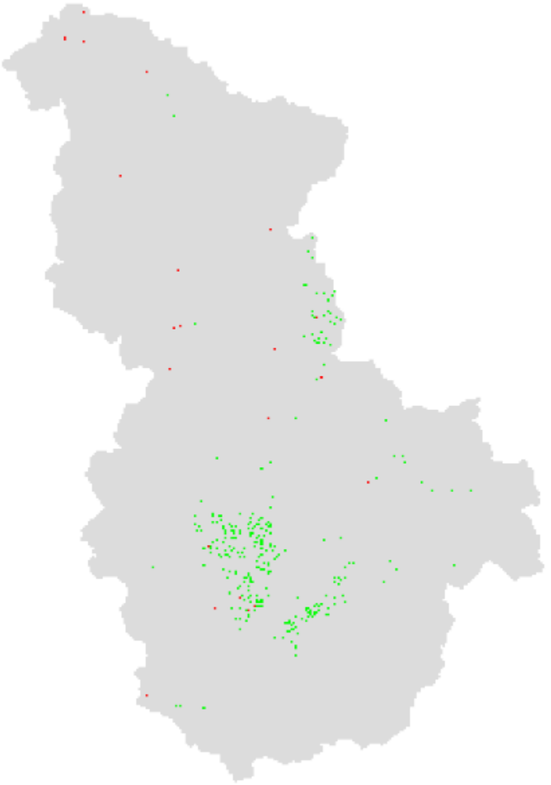
```python
lgbm_clf[4].get_params()
```

```
{'boosting_type': 'gbdt',
 'class_weight': None,
 'colsample_bytree': 0.649805253699943,
 'importance_type': 'split',
 'learning_rate': 0.0001324457034893535,
 'max_depth': 9,
 'min_child_samples': 20,
 'min_child_weight': 8,
 'min_split_gain': 0.0,
 'n_estimators': 4800,
 'n_jobs': -1,
 'num_leaves': 31,
 'objective': None,
 'random_state': None,
 'reg_alpha': 0.5968297802559932,
 'reg_lambda': 0.11582887397582241,
 'silent': True,
 'subsample': 1.0,
 'subsample_for_bin': 200000,
 'subsample_freq': 0,
 'colsample_bylevel': 0.7916400635068106,
 'colsample_bynode': 0.8833222413319739,
 'gamma': 0.559424285340915}
```

LightGBM의 Optuna 적용 모습

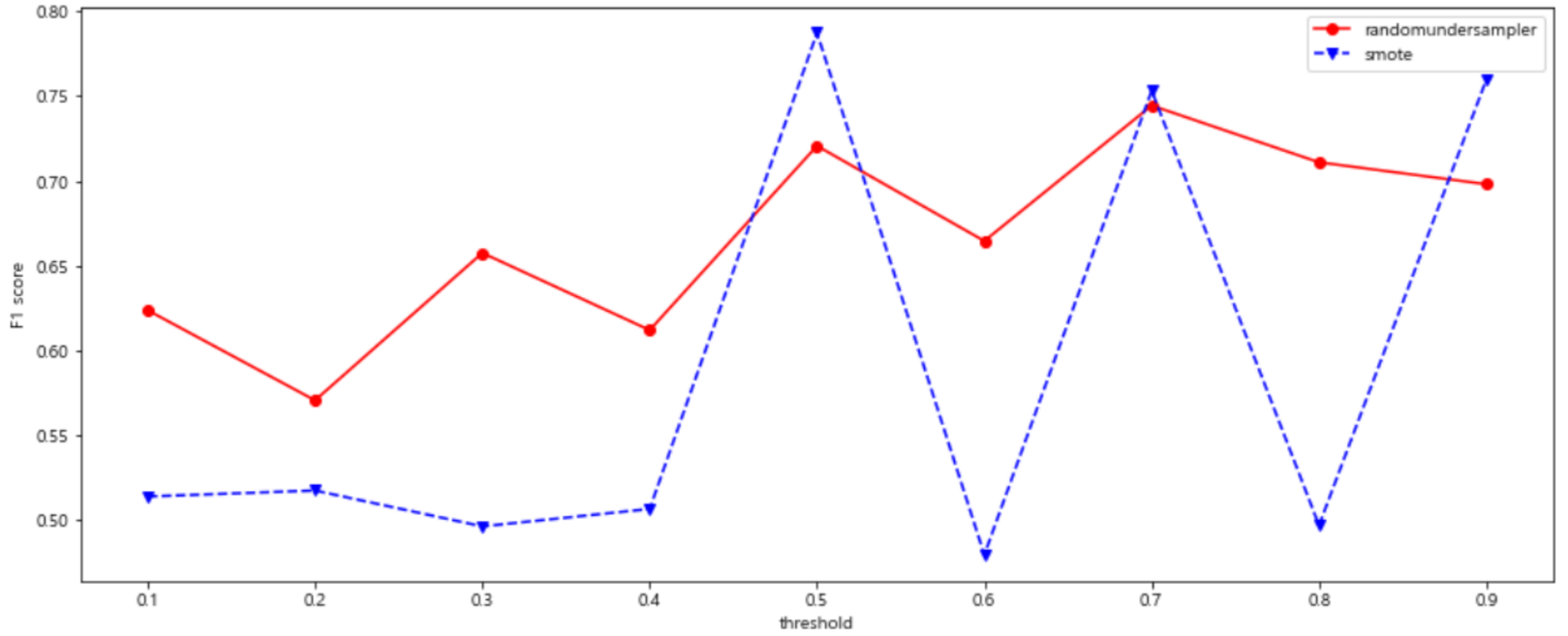
모델링 - Random Forest





임계치도 0.1~0.9로 조정하며 확인한 결과
F1 Score가 0.9가 넘는 모델들의 최적 입지는 총 22곳

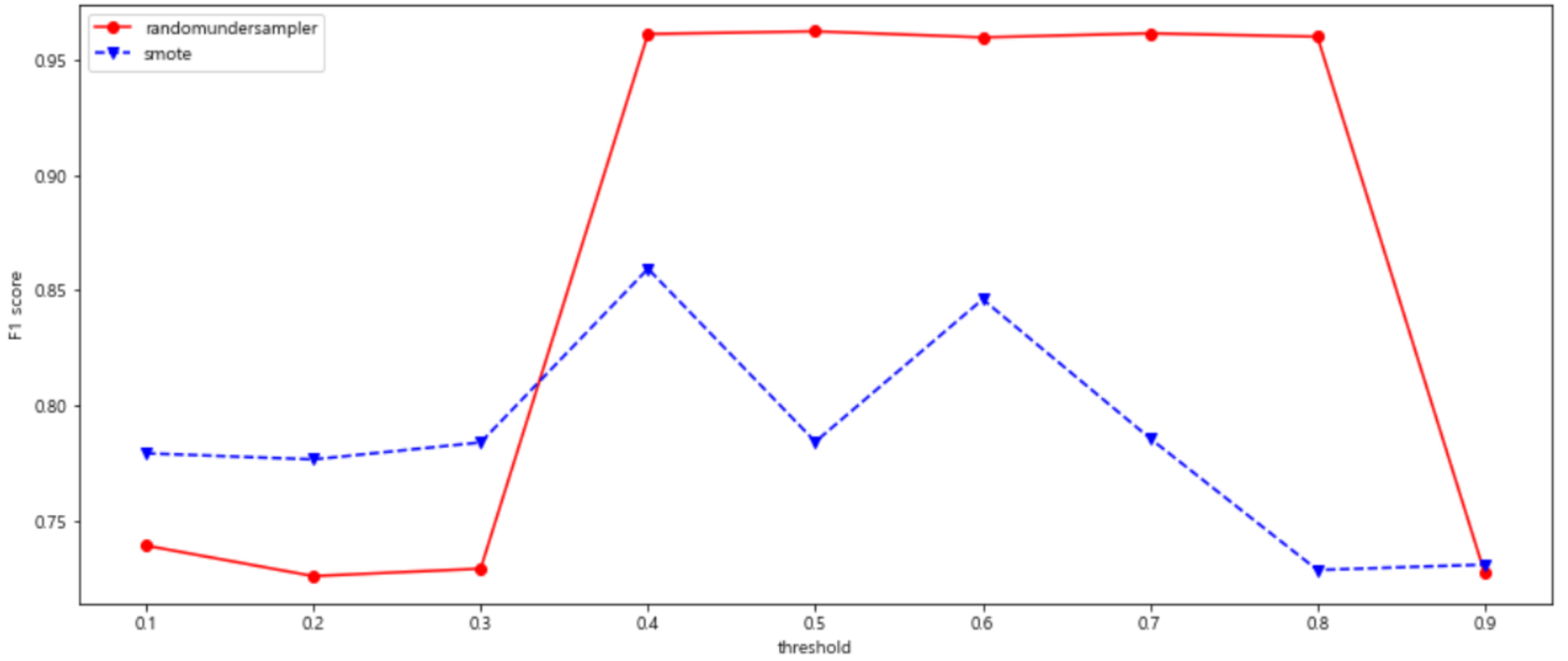
모델링 - LightGBM



	F1_score	TP	FP
0	0.623789	224	1237
1	0.570631	228	1422
2	0.657562	218	1116
3	0.612041	223	1276
4	0.720705	232	915
5	0.664611	229	1103
6	0.744493	238	840
7	0.711013	239	955
8	0.698091	232	992
9	0.513950	225	1612
10	0.517474	230	1605
11	0.496329	230	1677
12	0.506608	228	1640
13	0.787372	208	664
14	0.479589	237	1741
15	0.753304	214	786
16	0.496916	225	1670
17	0.759765	213	763

LightGBM은 F1 score가 0.9가 넘는 모델들이 없음
성능이 매우 낮아 입지 추천 수도 1,000개가 넘어가는 등 문제가 있어 제외

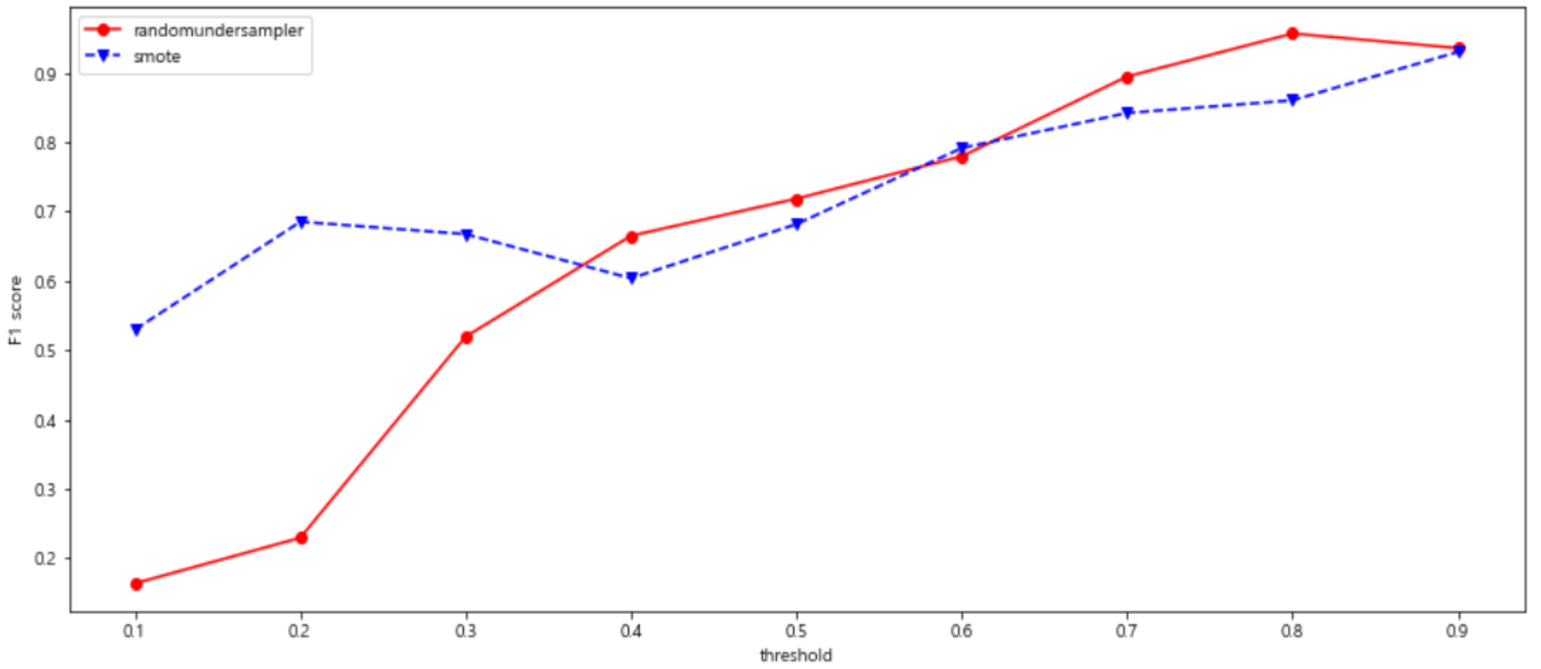
모델링 - CatBoost

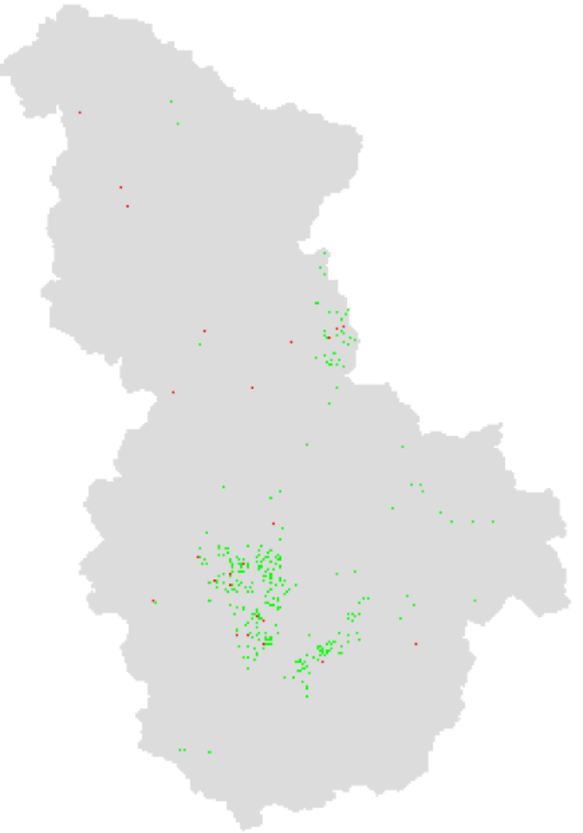


	F1_score	TP	FP
0	0.739501	241	860
1	0.726285	245	909
2	0.729515	243	896
3	0.960940	145	10
4	0.962115	143	4
5	0.959471	146	16
6	0.961233	145	9
7	0.959765	145	14
8	0.727460	243	903
9	0.779442	205	688
10	0.776799	217	709
11	0.784141	202	669
12	0.859325	184	395
13	0.784435	209	675
14	0.846109	194	450
15	0.785903	223	684
16	0.728928	245	900
17	0.731278	244	891

CatBoost는 언더샘플링 모델에서 성능 우수
총 19개의 최적 입지가 선택됨

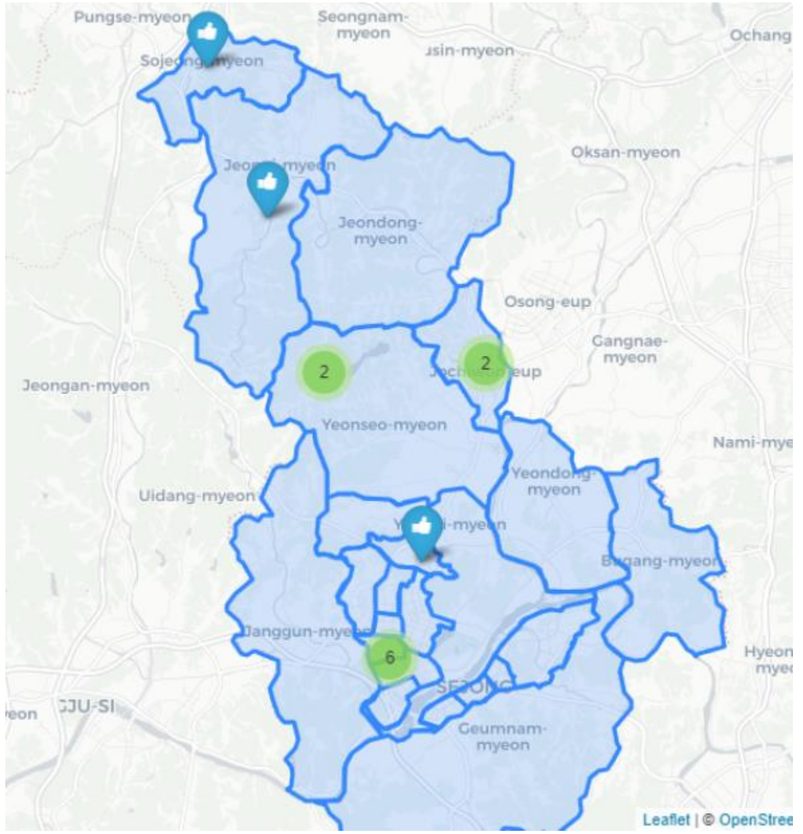
모델링 - Deep Learning (binary cross entropy)



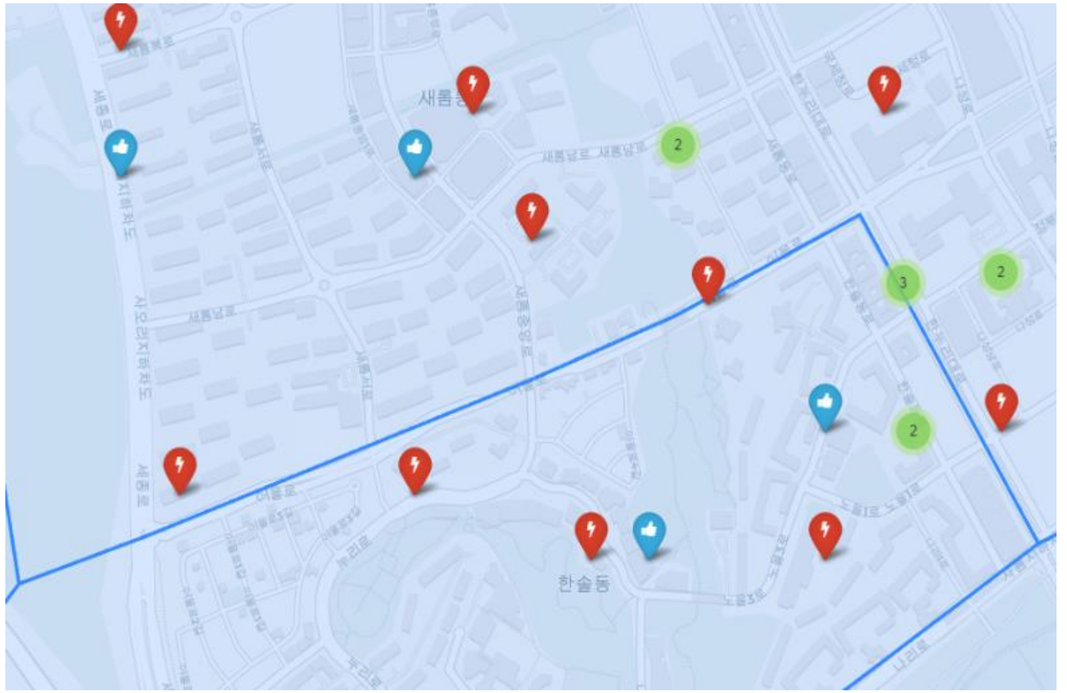


딥러닝의 경우 준수한 성능을 임계치가 상승할 수록 준수한 성능을 보임
최적 입지로 24곳을 선택

최종 결과



세종시 내 최적 입지 우선 순위로
13곳이 선정됨



기존 입지(빨강)과 최적 입지 추천 장소(파랑) 비교

04 기대 효과

공공기관

정책 달성을 위해 전기차 입지
선정시 최적의 위치 탐색 자료
로 활용 가능

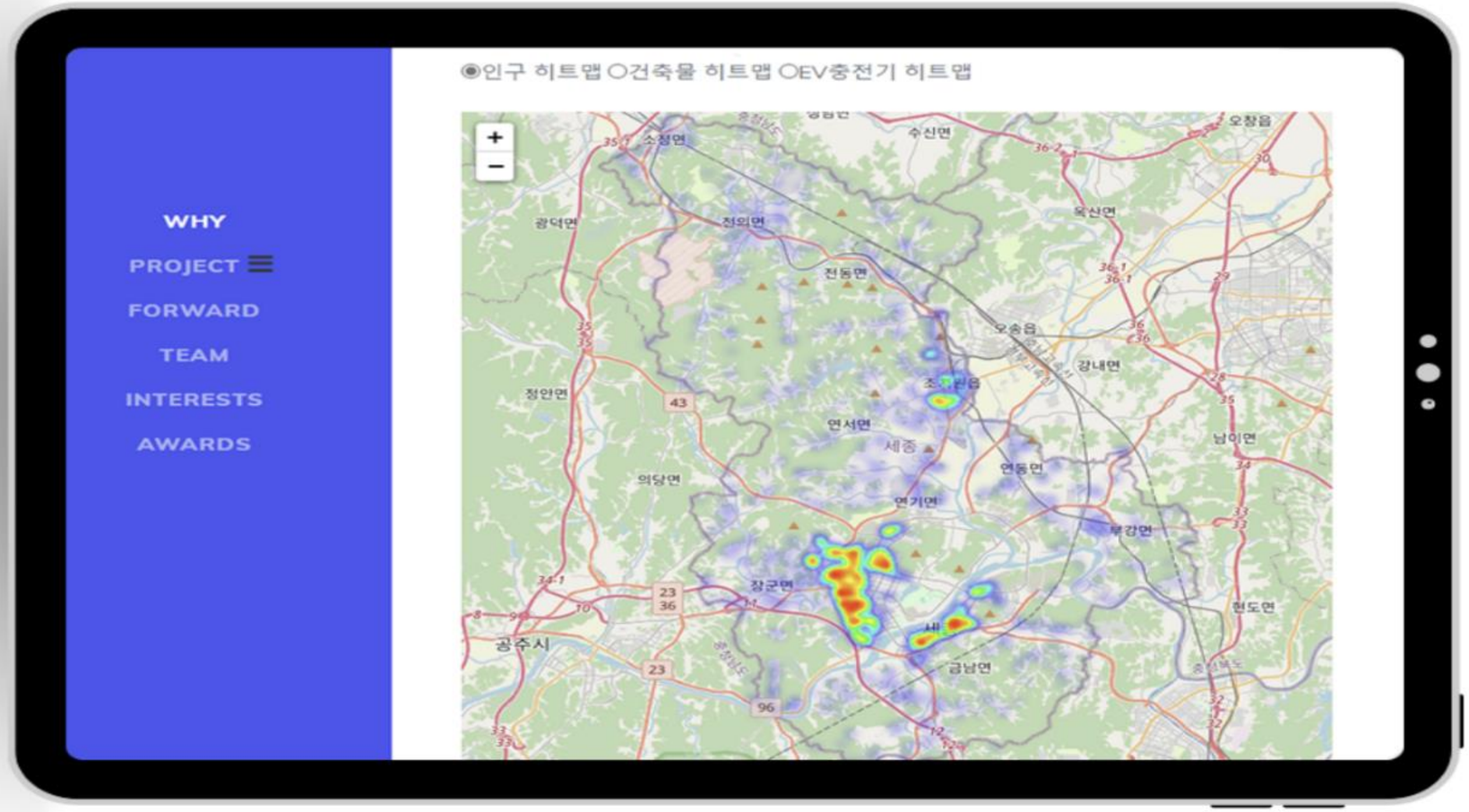
기업

전기차 충전소
이해관계자에게
최적 입지 선정의 솔루션 제공

세종 시민들

전기차를 이용하는
시민들의 편리성을 높임

04 웹 구현



* 클릭하면 사이트로 이동합니다.

05 참고 자료

전기차 시장 10년새 100배…올해 1000만대 넘본다
https://www.hani.co.kr/arti/science/future/1029869.html

전기차 및 충전기 보급·이용 현황 분석보고서_최종
https://new.kpx.or.kr/board.es?mid=a10502000000&bid=0045&act=view&list_no=51836&tag=&nPage=1

1인당 자동차 등록대수
https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_1YL20731

국토정보플랫폼 수직기준변환 KNGeoid18
http://map.ngii.go.kr/ms/mesrInfo/geoidIntro.do

전기차 충전소 모니터링 https://www.ev.or.kr/evmonitor#

전국주차장정보표준데이터 https://www.data.go.kr/data/15012896/standard.do

전국행정동경계 https://github.com/vuski/admdongkor

국토교통부_건축물대장정보 서비스
https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15044713

세종특별자치시 교통정보현황
http://tdata.sejong.go.kr:8080/ingecep/launcher/dashboard/

국토정보플랫폼 국토정보맵
http://map.ngii.go.kr/ms/map/NlipMap.do?tabGb=statsMap