

데이터로 같이, 가치 있게(With Value)!

데이터십 해커톤 제 5회

# 대구시 불법주정차 완화를 위한 최적 단속경로 제안

분석 결과보고서

참여조: 대구 7조

참여자: 한석준(조장), 이주엽

이채현, 전신빈, 최수빈,  
황채원

씨에스리 컨소시엄



kpc 한국생산성본부

## 목 차

1. 분석 개요	8
1.1. 분석 배경 및 개요	8
1.1.1. 배경	8
1.1.2. 개요	9
1.2. 분석 목적 및 방향	10
1.2.1. 분석 목적 및 방향	10
1.3. 분석 결과 활용 방안	10
1.3.1. 분석 결과 활용 방안	10
2. 분석 데이터	11
2.1. 분석 데이터 목록	11
2.1.1. 불법주정차 단속 데이터	11
2.1.2. 주변 상권 및 건물 데이터	11
2.1.3. 교통 데이터	11
2.1.4. 지역 경계 데이터	11
2.1.5. 인구 데이터	11
2.1.6. 불법주정차 금지 구역	11
2.1.7. CCTV 데이터	11
2.2. 데이터 상세 설명	12
2.1.1. 불법주정차 단속 데이터	12
2.1.2. 주변 상권 및 건물 데이터	12
2.1.3. 교통 데이터	15
2.1.4. 지역 경계 데이터	19
2.1.5. 인구 데이터	19
2.1.6. 불법주정차 금지 구역	19
2.1.7. CCTV 데이터	21

<b>2.3. 데이터 정제 방안</b>	23
2.3.1. 구군별 단속현황 데이터 전처리	23
2.3.2. 상권 데이터 분류	23
2.3.3. 건물 용도별 csv 파일 분할	23
2.3.4. 횡단보도, 건물 데이터 변환	24
2.3.5. 대구 행정동 경계 데이터 생성	24
2.3.6. csv 파일 Q-GIS에 레이어 추가 후 행정동별 데이터 분류	25
2.3.7. 교차로 교통량 데이터 전처리	25
2.3.8. 행정동별 데이터 결합	25
2.3.9. 중복된 변수 제거	26
2.3.10. 결측치가 많은 변수 삭제	26
<b>3. 분석 프로세스</b>	27
<b>3.1. 분석 프로세스</b>	27
3.1.1. 데이터 수집	27
3.1.2. 데이터 전처리	27
3.1.3. 분석	27
3.1.4. 활용	28
<b>3.2. 분석 내용 및 방법</b>	28
3.2.1. 회귀분석(Regression)	28
3.2.2. 군집분석(Clustering)	30
3.2.3. 분류분석(Classification)	31
<b>4. 분석 결과</b>	33
<b>4.1. 변수 선택</b>	33
4.1.1. 회귀분석	33
4.1.2. 비 계층적 군집분석	34
4.1.3. 계층적 군집분석	35
4.1.4. 분류 모델링	36
<b>4.2. 가중치 산정</b>	38
4.2.1. 가중치 산정 방법	38
4.2.2. 산정 결과	39

<b>4.3. 최적 경로 선정</b>	41
4.3.1. 총점 계산	41
4.3.2. 최적 경로 도출	42
<b>5. 활용 방안</b>	46
<b>5.1. 문제점 개선 방안</b>	46
5.1.1. 불법주정차 단속체계 정비의 가이드라인 제공	46
5.1.2. CCTV 추가 설치 필요	46
<b>5.2. 업무 활용 방안</b>	47
5.2.1. 지방자치단체 업무 활용	47
5.2.2. 표준 모델 활용	47
<b>6. 참고자료(Reference)</b>	48
<b>7. 부록</b>	52
<b>7.1. 추가 분석 결과</b>	52
7.1.1. 가중치 산정에 이용한 모델 평가 및 해석	52
7.1.2. 최적 경로 선정	56
<b>7.2. 주제설계를 위한 마인드맵</b>	58
<b>7.3. 분석 상세코드</b>	59
7.3.1. 단속 현황 전처리(2.3.1)	59
7.3.2. 좌표변환(2.3.1)	59
7.3.3. 교통량 결측치(2.3.7)	60
7.3.4. 다중회귀분석 단계적 선택법(4.1.1)	61
7.3.5. 비 계층적 군집분석(K-means)(4.1.2)	61
7.3.6. 계층 군집(Hierarchical Clustering)(4.1.3)	62
7.3.7. Pycaret(4.1.4 ~ 4.2)	63
7.3.8. 총점 계산(4.3.1)	64
7.3.9. 데이터 변환 및 오버샘플링(4.1.4)	66

## &lt;표 차례&gt;

[표 2-1] 구군별 불법주정차 단속현황 상세 설명	12
[표 2-2] 주요 상권 현황 상세 설명	12
[표 2-3] 지하철 주요시설 상세 설명	12
[표 2-4] 상가정보 상세 설명	13
[표 2-5] 상가정보 상세 설명	14
[표 2-6] 건물통합정보 상세 설명	14
[표 2-7] 건물통합정보 상세 설명	15
[표 2-8] 구군별 주차장 정보 상세 설명	15
[표 2-9] 구군별 주차장 정보 상세 설명	16
[표 2-10] 대구광역시 도로망 상세 설명	16
[표 2-11] 대구광역시 도로망 상세 설명	17
[표 2-12] 교차로 교통량 상세 설명	17
[표 2-13] 교차로 교통량 상세 설명	18
[표 2-14] 전국 행정동 경계 데이터 상세 설명	19
[표 2-15] 읍면동별 주민등록인구(2021년 10월 기준)	19
[표 2-16] 소화전 상세 설명	19
[표 2-17] 소화전 상세 설명	20
[표 2-18] 어린이보호구역 상세 설명	20
[표 2-19] 횡단보도 상세 설명	21
[표 2-20] 버스 노선 공간정보 상세 설명	21
[표 2-21] 시내버스 탑재형 불법주정차 단속 시스템 설치 현황 상세 설명	21
[표 2-22] 주차단속 카메라 정보 상세 설명	22
[표 4-1] 변수 선택 결과	37
[표 4-2] SHAP values 기반 가중치 산정을 위한 점수 및 순위	40
[표 4-3] SHAP values 기반 가중치 산정 결과	41

## &lt;그림 차례&gt;

[그림 1-1] 불법주정차 현황(2016~2021)	8
[그림 2-1] 구군별 불법주정차 단속현황 시각화(3개년 1,095건 이상)	23
[그림 2-2] 횡단보도 파일 변환 결과	24
[그림 2-3] 건물 파일 변환 결과	24
[그림 2-4] 대구 행정동 경계 shp 데이터 생성 결과	25
[그림 3-1] 분석 프로세스	27
[그림 3-2] 단순선형회귀 적합그래프	28
[그림 3-3] 다중선형회귀 적합그래프	29

[그림 3-4] 병합 군집 과정	31
[그림 3-5] Pycaret을 이용한 분류모델 산출	32
[그림 3-6] 제공 plot 종류	32
[그림 4-1] 비 계층적 군집분석 결과 시각화	34
[그림 4-2] 비 계층적 군집분석 실루엣 계수 시각화	35
[그림 4-3] 계층적 군집분석 결과	35
[그림 4-4] 변수중요도 Plot	36
[그림 4-5] SMOTE, fold 수 10일 때, 모델 매트릭스	39
[그림 4-6] 랜덤포레스트, SMOTE, fold 수 10일 때, SHAP values	39
[그림 4-7] 대구시 중구 순위 격자 시각화 결과 (전체 시각화 결과는 부록 첨부)	42
[그림 4-8] 대구시 중구 불법주정차 다발 구역 최적 단속경로	42
[그림 4-9] 대구시 동구 불법주정차 다발 구역 최적 단속경로	43
[그림 4-10] 대구시 서구 불법주정차 다발 구역 최적 단속경로	43
[그림 4-11] 대구시 남구 불법주정차 다발 구역 최적 단속경로	43
[그림 4-12] 대구시 북구 불법주정차 다발 구역 최적 단속경로	44
[그림 4-13] 대구시 수성구 불법주정차 다발 구역 최적 단속경로	44
[그림 4-14] 대구시 달서구 불법주정차 다발 구역 최적 단속경로	44
[그림 4-15] 대구시 달성군 불법주정차 다발 구역 최적 단속경로	45
[그림 4-16] 대구시 달성군 불법주정차 다발 구역 최적 단속경로	45
[그림 5-1] 대구시 고정형 CCTV 설치 현황	46
[그림 5-2] 불법주정차 다발 구역 부근 CCTV 설치 현황	47
[그림 6-1] 도로교통법 제32조	48
[그림 6-2] 도로교통법 제32조, 제33조, 제34조	49
[그림 6-3] 주정차 위반 단속	50
[그림 6-4] 불법주정차 단속 흐름도	50
[그림 6-5] 단속 권한 및 단속체계	51
[그림 6-6] 도로주정차선별 주정차 가능 여부	51
[그림 7-1] ADASYN, fold 5일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스	52
[그림 7-2] ADASYN, fold 5일 때 SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)	52
[그림 7-3] ADASYN, fold 5일 때, SHAP Values 및 변수중요도(Logistic Regression)	52
[그림 7-4] ADASYN, fold 10일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스	53
[그림 7-5] ADASYN, fold 10일 때 SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)	53
[그림 7-6] ADASYN, fold 10일 때, SHAP Values 및 변수중요도(Logistic Regression)	53
[그림 7-7] SMOTE, fold 5일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스	54
[그림 7-8] SMOTE, fold 5일 때, SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)	54
[그림 7-9] SMOTE, fold 5일 때, SHAP Values 및 변수중요도(Logistic Regression)	54

[그림 7-10] SMOTE, fold 10일 때, pycaret[이] 돌린 모든 모델들의 성능 매트릭스	55
[그림 7-11] SMOTE, fold 10일 때 SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)	55
[그림 7-12] SMOTE, fold 10일 때, SHAP Values 및 변수중요도(Logistic Regression)	55
[그림 7-13] 대구시 격자 순위 시각화 결과	56
[그림 7-14] 대구시 구별 불법주정차 최적 단속경로	57
[그림 7-15] 분석논리모형 마인드맵	58
[그림 7-16] 분석논리모형 마인드맵	58

## 1. 분석 개요

### 1.1. 분석 배경 및 개요

#### 1.1.1. 배경

□ 대구시 3대 전략사업 추진

2022년, 대구시는 3대 전략사업을 통한 시민 주차 편의 증진 정책을 추진 할 것을 발표했으나, 단순히 차량 대비 주차 확보율을 늘리는 것만으로는 불법주정차 문제 해결 불가능

\* 3대 전략사업: 주차시설 확충, 통합 주차 정보시스템 구축, 주차 수요 관리로 주 차 확보율 96.8%(2021년 기준)에서 98.6%로 확대 계획

□ 대구시 화물차 불법주정차 문제

최근, 화두 되는 대구시 화물차 불법주정차 문제를 포괄할 수 있는 효율적인 불법주정차 단속경로 도출

\* 2021년 말 기준 대구시 차고지 등록이 필요한 2.5 톤 이상 화물차량은 1만 5,000대, 주차 면수는 공영과 민영 차고지를 합쳐 1,390면 정도로 100대 중 9대 만 주차할 수 있어 91대는 불법주차를 하고 있으며, 이로 인하여 대형 차량이 시야를 가려 교통사고 위험이 큼.

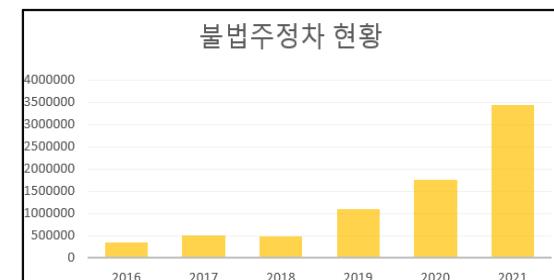
□ 매년 불법주정차 단속 건수 증가

대구시에서 발표한 불법주정차 단속 건수가 매년 큰 폭으로 늘어나고 있어 근본적 해결책 필요

2022년 불법주정차 단속 건수는 6월까지의 자료임에도 불구하고 2021년을 제외한 모든 연도의 단속 건수보다 많음

국민권익위원회에서 제공하는 민원 현황에 의하면, 2019년부터 불법주정차 단속 건수가 급격히 증가하는 경향

\* 2016년 347,459건, 2017년 500,739건, 2018년 475,073건, 2019년 1,088,343 건, 2020년 1,755,933건, 2021년 3,437,284건



[그림 1-1] 불법주정차 현황(2016~2021)

복잡한 불법주정차 민원신고 방법

불법주정차 신고 시, 5대 불법주정차의 경우 1분, 기타 불법주정차의 경우 5분 간격으로 신고 앱을 통해 사진을 두 번 제출해야 접수가 되는 형식으로 신고 절차가 복잡함

- \* 5대 불법주정차 금지 구역: 소방시설 주변 5m 이내, 교차로 모퉁이 5m 이내, 버스정류장 10m 이내, 횡단보도 위, 어린이 보호구역

불법주정차 단속 기준 부재

불법주정차 단속 차량 경로에 일정한 기준이 없고, 주로 민원인의 신고를 근거로 한 선제적이지 못한 후속 조치의 한계가 있으며, 각 구청 교통과마다 단속 기준에 편차가 있음

#### 1.1.2. 개요

- 불법주정차 단속현황, 주변 상권 및 건물, 주차구획 수 등의 데이터를 사용하여 3개년(2019~2021) 동안의 장소별, 구별 현황 파악 및 변화추이 분석, 단속 다발 구역 시각화
- SHAP values를 활용하여 양과 음의 가중치 산출
  - 하나의 모델이 아닌, 상위 n개 모델의 SHAP values를 모두 고려하여 포괄적으로 산출
    - \* SHAP values: 모델에 특정 변수가 있는 경우와 없는 경우를 비교하여 변수의 중요도를 계산, 변수를 공정하게 비교하기 위해 가능한 모든 순서로 수행하여 변수를 공정하게 비교하는 기법
- 가중치를 통해 총점을 계산하여 단속 필수구역 산출
  - 총점이 높은 구역을 단속 필수구역으로 선정
  - 구역별 총점 시각화를 통한 불법주정차 단속 필수구역 가시화
- 단속경로 최적화 문제 구성 및 지능형 알고리즘 적용
  - 단속 필수구역을 포괄할 수 있는 도로를 노드로 지정
  - 각 구청에서 출발하여 단속 필수구역을 최단으로 지나는 최적 경로 도출

## 1.2. 분석 목적 및 방향

### 1.2.1. 분석 목적 및 방향

- Q-GIS를 활용한 구별 불법주정차 다발 구간, 빈도수 시각화
- 주정차 위반 빈발구역 주변 건물 및 인프라, 도로 조사(거리뷰 등 이용)
- 다양한 변수를 고려하여 불법주정차 단속 및 민원 발생의 원인 규명 및 발생지역 식별, 불법주정차에 영향을 미치는 인자를 파악한 후 가중치 부여
- 가중치를 토대로 총점을 계산하는 식을 도출하여 주요 불법주정차 발생지역을 중심으로 한 최적 단속경로 제안
- 불법주정차 문제를 데이터를 기반으로 한 현황 및 원인 분석을 통해 불법주정차 완화 정책 제안

## 1.3. 분석 결과 활용 방안

### 1.3.1. 분석 결과 활용 방안

- 데이터를 기반으로 단속을 위한 구역을 설정하여 최적의 단속 노선 도출, 이를 차후 타지역에서도 활용할 수 있도록 표준 모델 개발
- 일정 기준 없이 민원 발생 후에 해결하던 임기응변식 처리에서 선제적 해결 방안 제공
- 현황 분석을 통해 대구시 특성에 맞는 주차정책과 단속체계 정비의 가이드라인 제공

## 2. 분석 데이터

### 2.1. 분석 데이터 목록

#### 2.1.1. 불법주정차 단속 데이터

- 구군별 불법주정차 단속현황

#### 2.1.2. 주변 상권 및 건물 데이터

- 주요 상권 현황
- 지하철 주요시설
- 상가정보
- 건물통합정보

#### 2.1.3. 교통 데이터

- 구군별 주차장 정보
- 대구광역시 도로망
- 교차로 교통량

#### 2.1.4. 지역 경계 데이터

- 전국 행정동 경계 데이터

#### 2.1.5. 인구 데이터

- 읍면동별 주민등록인구(2021년 10월 기준)

#### 2.1.6. 불법주정차 금지 구역

- 소화전
- 어린이보호구역
- 횡단보도
- 버스 노선 공간정보

#### 2.1.7. CCTV 데이터

- 시내버스 탑재형 불법주정차 단속시스템 설치 현황
- 주차단속 카메라 정보

### 2.2. 데이터 상세 설명

#### 2.1.1. 불법주정차 단속 데이터

[표 2-1] 구군별 불법주정차 단속현황 상세 설명

구분	변수명	상세 설명
1	위반일자	단속 일자
2	위반시간	단속 시간
3	위반장소명	단속 장소

#### 2.1.2. 주변 상권 및 건물 데이터

[표 2-2] 주요 상권 현황 상세 설명

구분	변수명	상세 설명
1	상권 번호	상권 번호
2	상권명	주요 상권이름
3	시도코드	대구광역시 코드
4	시도명	시도명
5	시군구코드	대구광역시 구군 코드
6	시군구명	대구광역시 구군명
7	상권좌표수	구군내 상권 내 좌표 개수
8	상권좌표	상권 내 좌표
9	데이터기준일자	기준일자

[표 2-3] 지하철 주요시설 상세 설명

구분	변수명	상세 설명
1	BASE_DATE	기준일자
2	ADM_DR_CD	읍면동 코드
3	OBJECTID	일련번호
4	읍면동	읍면동명
5	NUMPOINTS	동별 지하철 주요시설 위치 개수

[표 2-4] 상가정보 상세 설명

구분	변수명	상세 설명
1	상가업소번호	해당 업소 일련번호
2	상호명	상호명
3	지점명	지점명
4	상권업종대분류코드	대분류코드
5	상권업종대분류명	업종대분류명
6	상권업종중분류코드	중분류코드
7	상권업종중분류명	업종중분류명
8	상권업종소분류코드	소분류코드
9	상권업종소분류명	업종소분류명
10	표준산업분류코드	표준분류코드
11	표준산업분류명	표준산업명
12	시도코드	시도코드
13	시도명	시도명
14	시군구명	시군구명
15	행정동코드	각 시별 행정동 코드
16	행정동명	각 시별 행정동명
17	법정동코드	각 시별 법정동 코드
18	법정동명	각 시별 법정동명
19	지번코드	지번 코드
20	대지구분코드	대지 구분
21	대지구분명	대지 구분명
22	지번본번지	해당 지번주소 본번지
23	지번부번지	해당 지번주소 부번지
24	지번주소	전체 지번주소
25	도로명코드	도로명주소코드
26	도로명	해당 주소 도로명
27	건물본번지	해당 도로명주소 건물 본번지
28	건물부번지	해당 도로명주소 건물 부번지

[표 2-5] 상가정보 상세 설명

구분	변수명	상세 설명
29	건물관리번호	해당상가 건물관리번호
30	건물명	상가 입점 건물명
31	도로명주소	도로명주소
32	구우편번호	변경 전 우편번호
33	신우편번호	현재 우편번호
34	동정보	건물 동정보
35	층정보	건물 층정보
36	호정보	건물 호정보
37	경도	경도 좌표
38	위도	위도 좌표

[표 2-6] 건물통합정보 상세 설명

구분	변수명	상세 설명
1	UFID	UFID
2	BLD_NM	건물명칭
3	DONG_NM	동명칭
4	GRND_FLR	건물통합 지상층수
5	UGRND_FLR	건물통합 지하층수
6	PNU	토지코드
7	ARCHAREA	건물통합건축면적
8	TOTALAREA	건물통합 연면적
9	PLATAREA	건물통합대지면적
10	HEIGHT	건물통합 높이
11	STRCT_CD	구조
12	USABILITY	용도
13	BC_RAT	건물통합 건폐율
14	VL_RAT	건물통합 용적률

[표 2-7] 건물통합정보 상세 설명

구분	변수명	상세 설명
15	BLDRGST_PK	건축물대장 PK
16	USEAPR_DAY	승인일자
17	REGIST_DAY	데이터생성 변경일자
18	GB_CD	구분
19	VOL_BD_YN	위반건축물
20	GEOIDN	참조체계연계키
21	BLDG_PNU	건물 필지고유번호
22	BLDG_PNU_YN	건물 필지고유번호 유무
23	BLD_UNLICENSE	건물 무허가 여부
24	BD_MGT_SN	도로명주소건물관리번호
25	SGG_OID	원천오브젝트ID
26	COL ADM_SECT_CD	원천시군구코드
27	OBJECTID	OBJECTID
28	SHAPE	도형ID

## 2.1.3. 교통 데이터

[표 2-8] 구군별 주차장 정보 상세 설명

구분	변수명	상세 설명
1	주차장관리번호	주차장 관리 번호
2	주차장명	주차장명
3	주차장구분	주차장 구분(공영, 민영)
4	주차장유형	주차장 유형(노상, 노외)
5	소재지도로명주소	주차장 도로명 주소
6	소재지지번주소	주차장 지번 주소
7	주차구획수	주차 면수
8	급지구분	1: 도심지역, 2: 도심지역 중 1급지제외지역, 3: 역 세권, 4: 상위급지 제외 노상 및 노외주차장
9	부제시행구분	부제 시행 여부

[표 2-9] 구군별 주차장 정보 상세 설명

구분	변수명	상세 설명
10	운영요일	운영 요일
11	평일운영시작시각	평일 운영 시작 시각
12	평일운영종료시각	평일 운영 종료 시각
13	토요일운영시작시각	토요일 운영 시작 시각
14	토요일운영종료시각	토요일 운영 종료 시각
15	공휴일운영시작시각	공휴일 운영 시작 시각
16	공휴일운영종료시각	공휴일 운영 종료 시각
17	요금정보	요금정보(무료, 유료)
18	주차기본시간	주차 기본 시간
19	주차기본요금	주차 기본 요금
20	추가단위시간	추가 단위 시간
21	추가단위요금	추가 단위 당 요금
22	1일주차권요금적용시간	1일 주차권 요금 적용 시간
23	1일주차권요금	1일 주차권 요금
24	월정기권요금	월정기권 요금
25	결제방법	결제방법(현금, 신용카드)
26	특기사항	할인정보
27	관리기관명	주차장 관리기관명
28	전화번호	주차장 관리기관 전화번호
29	위도	주차장 위치(위도)
30	경도	주차장 위치(경도)
31	데이터기준일자	데이터 수집 일자

[표 2-10] 대구광역시 도로망 상세 설명

구분	변수명	상세 설명
1	ALWNC_DE	부여일자
2	ALWNC_RESN	부여사유
3	BSI_INT	기초간격

[표 2-11] 대구광역시 도로망 상세 설명

구분	변수명	상세 설명
4	ENG_RN	영문도로명
5	MVMN_DE	이동일자
6	MVMN_RESN	이동사유
7	MVM_RES_CD	이동사유코드
8	NTFC_DE	고시일자
9	OPERT_DE	작업일시
10	RBP_CN	기점
11	RDS_DPN_SE	도로구간종속구분
12	RDS_MAN_NO	도로구간일련번호
13	REP_CN	종점
14	RN	도로명
15	RN_CD	도로명코드
16	ROAD_BT	도로폭
17	ROAD_LT	도로길이
18	ROA_CLS_SE	도로위계기능구분
19	SIG_CD	시군구코드
20	WDR_RD_CD	광역도로구분코드

[표 2-12] 교차로 교통량 상세 설명

구분	변수명	상세 설명
1	조사지점	교차로명
2	계 (대/6h)	일정 시간(6h) 대비 전체 차량 이동 대수
3	PCU 합계 (PCU/6h)	일정 시간(6h) 대비 승용차 환산대수 합계
4	첨두환산 합계 (대/6h)	일정 시간(6h) 교통량을 첨두시간계수(PHF)로 나눈 교통량 합계
5	이륜 (대/6h)	일정 시간(6h) 대비 이륜차 이동 대수
6	승용차 (대/6h)	일정 시간(6h) 대비 승용차 이동 대수
7	버스 소형 (대/6h)	일정 시간(6h) 대비 버스 소형 이동 대수
8	버스 중형 (대/6h)	일정 시간(6h) 대비 버스 중형 이동 대수

[표 2-13] 교차로 교통량 상세 설명

구분	변수명	상세 설명
9	버스 대형 (대/6h)	일정 시간(6h) 대비 버스 대형 이동 대수
10	화물차 소형 (대/6h)	일정 시간(6h) 대비 화물차 소형 이동 대수
11	화물차 중형 (대/6h)	일정 시간(6h) 대비 화물차 중형 이동 대수
12	화물차 대형 (대/6h)	일정 시간(6h) 대비 화물차 대형 이동 대수
13	토요일운영시작시각	토요일 운영 시작 시각
14	토요일운영종료시각	토요일 운영 종료 시각
15	공휴일운영시작시각	공휴일 운영 시작 시각
16	공휴일운영종료시각	공휴일 운영 종료 시각
17	요금정보	요금정보(무료, 유료)
18	주차기본시간	주차 기본 시간
19	주차기본요금	주차 기본 요금
20	추가단위시간	추가 단위 시간
21	추가단위요금	추가 단위 당 요금
22	1일주차권요금적용시간	1일 주차권 요금 적용 시간
23	1일주차권요금	1일 주차권 요금
24	월정기권요금	월정기권 요금
25	결제방법	결제 방법(현금, 신용카드)
26	특기사항	할인정보
27	관리기관명	주차장 관리기관명
28	전화번호	주차장 관리기관 전화번호
29	위도	주차장 위치(위도)
30	경도	주차장 위치(경도)
31	데이터기준일자	데이터 수집 일자

## 2.1.4. 지역 경계 데이터

[표 2-14] 전국 행정동 경계 데이터 상세 설명

구분	변수명	상세 설명
1	BASE_DATE	기준일자
2	ADM_DR_CD	읍면동 코드
3	ADM_DR_NM	읍면동 이름
4	OBJECTID	일련번호

## 2.1.5. 인구 데이터

[표 2-15] 읍면동별 주민등록인구(2021년 10월 기준)

구분	변수명	상세 설명
1	행정구역(동읍면)	시군구, 동읍면
2	총인구수	총인구수
3	남자인구수	남자인구수
4	여자인구수	여자인구수

## 2.1.6. 불법주정차 금지 구역

[표 2-16] 소화전 상세 설명

구분	변수명	상세 설명
1	시설번호	소화전 번호
2	시설유형코드	1=지상, 2=지하, 3=급수탑, 4=저수조)
3	시도명	소화전이 설치된 시도명
4	시군구명	소화전이 설치된 시군구명
5	시군구코드	소화전이 설치된 시군구 코드
6	소재지도로명주소	소화전의 도로명주소
7	소재지지번주소	소화전의 지번주소
8	위도	소화전의 위치(위도)
9	경도	소화전의 위치(경도)
10	상세위치	소화전의 상세위치

[표 2-17] 소화전 상세 설명

구분	변수명	상세 설명
11	안전센터명	소화전 담당 안전센터
12	보호틀유무	보호틀 여부(Y, N)
13	사용가능여부	소화전 사용 가능 여부
14	설치연도	소화전 설치 연도
15	배관깊이	소화전의 배관 깊이
16	출수압력	소화전의 출수 압력
17	배관지름	소화전의 배관 지름
18	관할기관명	관할 소방서명
19	관할기관전화번호	관할 소방서 전화번호
20	데이터기준	데이터 기준 일자

[표 2-18] 어린이보호구역 상세 설명

구분	변수명	상세 설명
1	시설종류	어린이집, 유치원, 초등학교, 특수학교, 학원
2	대상시설명	시설명
3	소재지도로명주소	도로명주소
4	소재지지번주소	지번주소
5	위도	어린이보호구역 위치(위도)
6	경도	어린이보호구역 위치(경도)
7	관리기관명	관리기관명(구청)
8	관할경찰서	관할 경찰서명
9	CCTV 설치여부	CCTV 설치여부(Y, N)
10	CCTV 설치대수	CCTV 설치대수
11	보호구역도로폭	어린이 보호구역 도로폭
12	데이터기준	데이터 수집 기준 일자
13	제공기관코드	제공기관코드
14	제공기관명	제공기관명

[표 2-19] 횡단보도 상세 설명

구분	변수명	상세 설명
1	UQFTID_ID	번호
2	CNSTC_SE	분류
3	CNSTC_DE	설치일자

[표 2-20] 버스 노선 공간정보 상세 설명

구분	변수명	상세 설명
1	gid	번호
2	app_bg_dt	버스정류장 설치 일자
3	app_ed_dt	버스정류장 폐쇄 일자
4	bs_id	버스정류장 아이디
5	bs_nm	버스정류장명
6	bs_eng_nm	버스정류장 영문명
7	ngis_x_pos	경도 좌표
8	ngis_y_pos	위도 좌표
9	node_id	노드 아이디
10	node_k_cd	노선으로 연결된 링크의 수
11	route_no	해당 버스정류장의 운행 버스명
12	dataconnar	지역구분(GS:경산, DG:대구, CG:칠곡)

### 2.1.7. CCTV 데이터

[표 2-21] 시내버스 탑재형 불법주정차 단속 시스템 설치 현황 상세 설명

구분	변수명	상세 설명
1	버스노선	버스명
2	차번호	차량번호
3	버스회사	버스회사명
4	노선	버스 노선

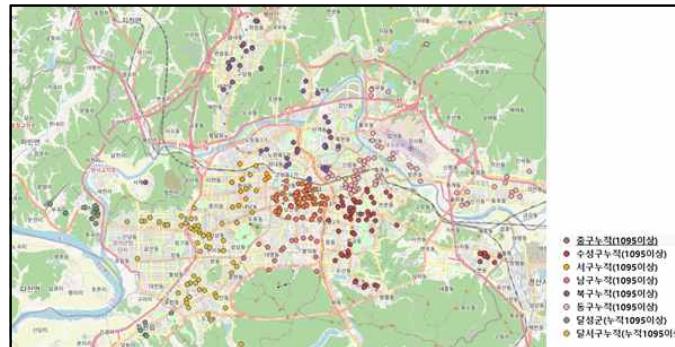
[표 2-22] 주차단속 카메라 정보 상세 설명

구분	변수명	상세 설명
1	공간정보	구군 코드를 포함한 공간정보
2	설치위치	대략적 설치 위치 구분
3	경도	경도 좌표
4	위도	위도 좌표
5	구군코드	구군별 코드
6	구군명	대구광역시 구, 군
7	단속카메라단속여부	단속카메라 여부
8	평일단속시작시간	평일 단속 시작 시각
9	평일단속종료시간	평일 단속 종료 시각
10	토요일단속시작시간	토요일 단속 시작 시각
11	토요일단속종료시간	토요일 단속 종료 시각
12	공휴일단속시작시간	공휴일 단속 시작 시각
13	공휴일단속종료시간	공휴일 단속 종료 시각
14	주정차단속비고	촬영 시간 간격 및 토, 일, 공휴일 단속정보

## 2.3. 데이터 정제 방안

### 2.3.1. 구군별 단속현황 데이터 전처리

- 문자열의 날짜 변수를 datetime64 형태로 변환하여 3개년('19~'21) 추출
- 단속 장소 기준으로 누적 수 value\_counts()
- 하루에 1번 이상 단속 지점 추출(3개년 동안의 단속 건수 1,095건 이상)
- 스프레드시트의 Geocoding 및 Python 내의 geopy를 활용하여 좌표 추출
- 추출된 좌표를 Q-GIS를 통해 시각화



[그림 2-1] 구군별 불법주정차 단속현황 시각화(3개년 1,095건 이상)

### 2.3.2. 상권 데이터 분류

- 변수 '상권업종대분류명'을 기준으로 데이터 분류
  - \* 관광여가오락(N), 부동산(L), 생활서비스(F), 소매(D), 숙박(O), 스포츠(P), 음식(Q), 학문교육(R)
- 행정동별 상권 수를 카운트 및 총합하여 상권업종별 칼럼 생성

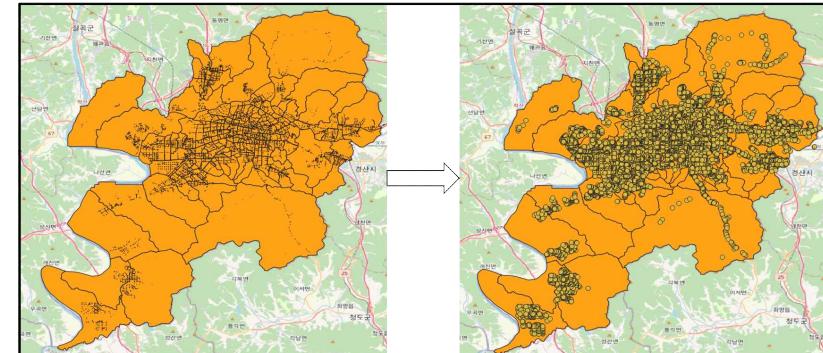
### 2.3.3. 건물 용도별 csv 파일 분할

- 건축법에서 정의한 용도 분류 기준에 따라 csv 파일로 저장

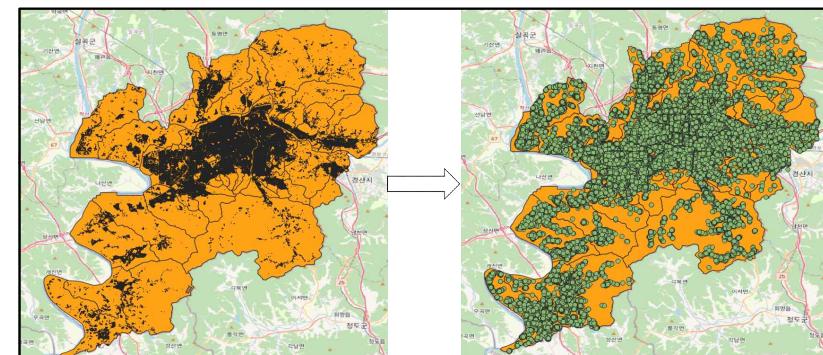
\* 단독주택, 공동주택, (1, 2종)근린생활, 문화 및 접회, 종교, 판매, 운수, 의료, 교육연구, 노유자, 수련, 운동, 업무, 숙박, 위락, 공장, 창고, 위험물 저장 및 처리, 자동차 관련, 동물 및 식물 관련, 자원순환 관련, 교정 및 군사, 방송통신, 발전, 묘지 관련, 관광 휴게, 장례식장 및 야영장으로 분류

### 2.3.4. 횡단보도, 건물 데이터 변환

- 횡단보도 레이어, 건물 레이어 추가
- Q-GIS 속성테이블 > 필드계산기 사용하여 좌표 변수 추가
- 경도: x(centroid(\$geometry))
- 위도: y(centroid(\$geometry))
- 속성테이블 csv파일 변환 후 레이어 추가하여 변환 결과 확인



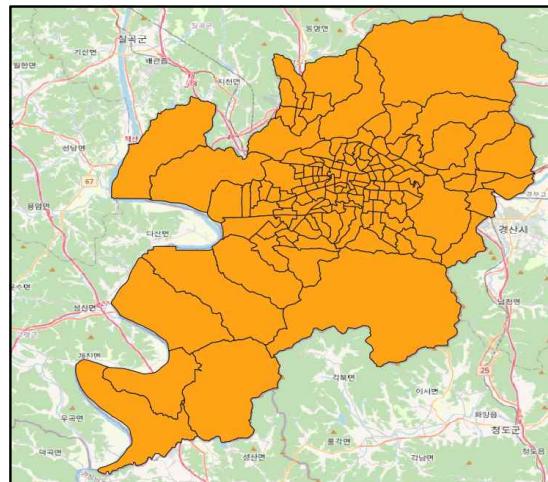
[그림 2-2] 횡단보도 파일 변환 결과



[그림 2-3] 건물 파일 변환 결과

### 2.3.5. 대구 행정동 경계 데이터 생성

- Q-GIS에 전국 행정동 경계 레이어(LSMD\_ADMIN\_SECT\_UMD\_27.shp) 추가
- 대구 경계와 붙어있는 지역 폴리곤 삭제(객체 선택을 쉽게 하기 위함)
- [사용자가 직접 그려 객체 선택] 사용하여 대구 지역 선택
- 선택한 객체를 다른 이름으로 저장 후 레이어 추가



[그림 2-4] 대구 행정동 경계 shp 데이터 생성 결과

## 2.3.6. csv 파일 Q-GIS에 레이어 추가 후 행정동별 데이터 분류

- [폴리곤에 포함하는 포인트 개수 계산] 사용하여 행정동별 개수 카운트
  - \* 폴리곤: 대구 행정동 경계
  - \* 포인트: 단속지점개수, 어린이보호구역, 주차장(가중치: 주차구획수), 버스정류장, 지하철주요시설, 소화전, 횡단보도, 건물(용도별), 상권(분류별)
- 포인트별로 csv파일 저장

## 2.3.7. 교차로 교통량 데이터 전처리

- 교통량 결측치 대체하기 위해 '행정동명'을 라벨 인코딩
- 전처리 전인 '건물'변수와 인코딩하기 전 '행정동명' 변수 삭제
- 교통량 결측의 패턴이 MAR(무작위 결측)이라고 판단
- MICE(Multivariate Imputation By Chained Equations) 사용
- Round robin 방식을 반복하여 결측값을 회귀(선형회귀), 다중 대치하는 방식

## 2.3.8. 행정동별 데이터 결합

- 행정동 기준 오름차순 정렬하여 행정동별로 분류한 데이터 결합
- 최종파일: 결합파일\_1095.csv

## 2.3.9. 중복된 변수 제거

- 의미가 중복되는 변수 제거
  - \* '교육연구시설' 변수는 '상권\_학문교육' 변수에 포함되어 '교육연구시설' 변수 삭제
  - \* '운동시설' 변수는 '상권\_스포츠' 변수에 포함되어 '운동시설' 변수 삭제
- 단독주택, 공동주택, 다세대주택은 '주택' 변수로 통합

## 2.3.10. 결측치가 많은 변수 삭제

- 결측치가 많거나, 0건인 행이 많은 변수 삭제

\* 생활편익시설(2004), 소매점(3001), 공공시설(3100), 파출소(3103), 판매시설(7000), 수련시설(12000), 위락시설(16000), 창고(18001), 위험물저장및처리시설(19000), 자동차관련시설(20000), 동식물관련시설(21000), 분뇨쓰레기처리시설(22000), 교정및군사시설(23000), 방송통신시설(24000), 발전시설(25000), 묘지관련시설(26000), 관광휴게시설(27000), 가설건축물(28000), 자원순환관련시설(30000), 근린생활시설(Z3000), 문화및집회시설(Z5000), 판매및영업시설(Z6000), 교육연구및복지시설(Z8000), 공공용시설(Z9000), 대형마트 변수 삭제

\* (): 건물용도코드

### 3. 분석 프로세스

#### 3.1. 분석 프로세스



##### 3.1.1. 데이터 수집

- 불법주정차 관련 데이터(주정차 금지 구역 등) 수집
- 교통유발시설(건물, 상권)과 교통량 데이터 수집
- CCTV 현황 데이터 수집

##### 3.1.2. 데이터 전처리

- 분석 가능하도록 데이터 정제 및 통합
- 범주형 데이터 수치화
- 행정동별로 수치화
- 종속변수의 불균형을 완화하기 위해 오버샘플링(SMOTE, ADASYN)
- 결측치 처리(MICE)
- 좌표계 변환

##### 3.1.3. 분석

- 중요 변수 선택 및 가중치 부여
- 단속 필수 지점 파악 후 최적 경로 도출을 위한 모델 적합 단계

#### 3.1.4. 활용

- 부여한 가중치로 총점을 계산하여 파악한 단속 필수 지점과 최적 경로 시작화

#### 3.2. 분석 내용 및 방법

##### 3.2.1. 회귀분석(Regression)

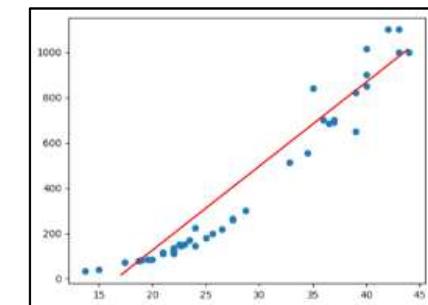
- 변수 중 하나를 종속변수(y)로 나머지 변수를 독립변수(x)로 하여 변수들이 서로 상관관계를 가질 때, 독립변수가 변화함에 따라 종속변수가 어떻게 변화하는가를 규명하는 분석 방법

- 선형회귀(Linear Regression): 가장 일반적인 유형, 변수들이 서로 선형적으로 연결되어 있을 경우 사용되며 독립변수의 수에 따라 단순선형회귀, 다중선형회귀로 구분

- \* 단순선형회귀(Simple Linear Regression):  
독립변수가 1개인 선형회귀분석

독립변수에 곱해지는 w값을 가중치(weight), 상수항에 해당하는 b를 편향(bias)이라 할 때, 추정된 회귀식은 다음과 같음

$$y = wx + b$$



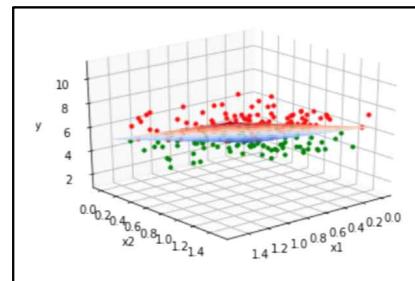
[그림 3-2] 단순선형회귀 적합 그래프

- \* 다중선형회귀(Multiple Linear Regression):

독립변수가 2개 이상인 선형회귀분석으로, 2차 이상의 차원에서 정의된 데이터들을 가장 잘 표현하는 평면이나 초평면을 찾는 분석 방법  
추가적인 독립변수를 도입하여 오차항의 값을 줄일 수 있고, 편향을 제거한다는 점에서 단순회귀분석의 단점 극복 가능

독립변수에 곱해지는 w값을 가중치(weight), 상수항에 해당하는 b를 편향(bias)이라 할 때, 추정된 회귀식은 다음과 같음

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$



[그림 3-3] 다중선형회귀 적합 그래프

- 로지스틱 회귀(Logistic Regression): 종속변수가 이산 값일 경우 S자형 곡선을 사용하여 종속변수와 독립변수 사이의 관계 표시
- 렉지 회귀(Ridge Regression): L2-Norm을 사용한 회귀로 일반적으로 영향을 거의 미치지 않는 특성에 대하여 0에 가까운 가중치 부여
- 라쏘 회귀(Lasso Regression): L1-Norm을 사용한 회귀로, 특성값의 개수가 매우 낮다면 0으로 수렴하게 하여 특성을 삭제하여 편향 증대, 과적합 방지
- 다항 회귀(Polynomial Regression): 독립변수가 여러 개인 다중 선형 회귀와 비슷하게 작동하지만, 비선형 곡선을 사용

#### □ 회귀모형 적합도 판단

- 분산분석(ANOVA)을 통해 회귀모형의 적합도를 판단할 수 있으며 F 통계량을 사용
  - \* F-분포: 정규분포를 이루는 모집단에서 독립적으로 추출한 표본들의 분산 비율이 나타내는 연속확률분포
  - \* 결정계수: 종속변수의 분산 중에서 독립변수로 설명되는 비율, 결정계수가 클수록 예측값과 실제값의 차이가 작아짐

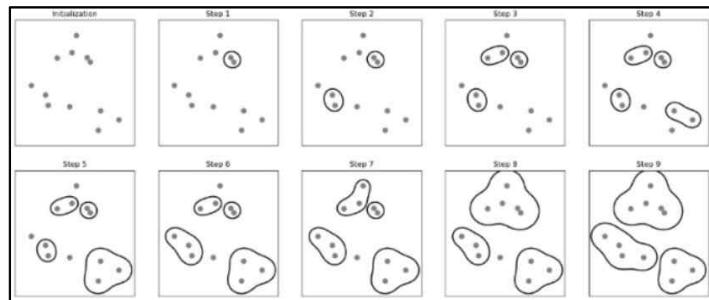
$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

#### □ 회귀계수 유의성 검정

- 추정된 각 회귀계수가 유의한지 t-검정을 통해 확인할 수 있음
  - \* t-검정: 두 집단 간의 평균을 비교하는 모수적 통계 방법으로, 표본이 정규성, 등분산성, 독립성 등을 만족할 경우에 적용 가능한 방법

#### 3.2.2. 군집분석(Clustering)

- 각 개체의 유사성을 측정하여 높은 대상 집단을 분류하고, 군집에 속한 개체들의 유사성과 서로 다른 군집에 속한 개체 간의 상이성을 규명하는 통계 분석 방법
- 개체들을 분류(classification)하기 위한 기준이 없는 상태에서 주어진 데이터의 속성값들을 고려해 유사한 개체끼리 그룹(클러스터)화하는 방법
- 유형: 비 계층적(Non-Hierarchical Clustering), 계층적(Hierarchical Clustering)
  - \* 비 계층적 군집분석: 계층을 두지 않고 그룹화를 할 유사도 측정 방식에 따라 최적의 그룹을 찾아가는 방법(k-means)
  - \* 계층적 군집분석: 여러 개의 군집 중에서 가장 유사도가 높은 혹은 거리가 가까운 군집 두 개를 선택하여 하나로 합치면서 군집 개수를 줄여가는 방법(병합 군집)
- 군집분석의 유사성 계산 척도(거리, 유사성)
  - 거리: 유클리드, 마할라노비스, 민코우스키, 맨하탄거리
  - 유사성: 일반적으로 변수들 사이의 상관계수 사용하며 식은 아래와 같음
 
$$* s_{ij} = \sqrt{\frac{\sum_{k=1}^p (X_{ik} - \bar{X}_{i.})(X_{jk} - \bar{X}_{j.})}{\sqrt{\sum_{k=1}^p (X_{ik} - \bar{X}_{i.})^2} \sqrt{\sum_{k=1}^p (X_{jk} - \bar{X}_{j.})^2}}} \text{ 여기서, } \bar{X}_{i.} = \frac{1}{p} \sum_{k=1}^p X_{ik}$$
- k-means: 각 군집의 평균(mean)을 활용하여 k개의 군집으로 묶는 것으로 k-nn과는 상이한 알고리즘
  - 각 그룹 내의 데이터들의 변동량이 최소가 되도록 함
  - $x_{kij}$ 를 k- 번째 그룹에 속한 i-행, j-열 자료라고 하고  $m_{kj}$ 를 k-번째 그룹의 i열 평균이라고 할 때, 아래 식을 최소화
 
$$* \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^p (x_{kij} - m_{kj})^2$$
- 병합 군집(Agglomerative Clustering) : 각각의 데이터 포인트를 하나의 클러스터로 지정하고, 지정된 개수의 클러스터가 남을 때까지 가장 비슷한 두 클러스터를 합쳐 나가는 알고리즘
  - 옵션: ward, average, complete
  - k-means와 달리 클러스터의 개수를 사전에 정하지 않고 사용



[그림 3-4] 병합 군집 과정

- 본 분석에서는 계층, 비 계층적 군집 모두 시도해보았으나 실루엣 계수가 유의미하다고 판단하기 어려워 기각

### 3.2.3. 분류분석(Classification)

- 주어진 입력변수에 근거하여 범주형 반응변수를 예측하는 분석
- 군집화와 유사하여 혼동할 수 있으나, 분류분석은 각 그룹이 정의되어있다는 점에서 차이가 있음
- 기법: 로지스틱 회귀, 의사결정나무, CART, 베이지안 분류, 인공신경망, SVM, K-NN 등
- 지표: 이항편차, 혼동행렬(정밀도, 재현율, FPR, 민감도, 정확도, F1 score, ROC곡선, AUC(ROC곡선 아래의 영역))
- 본 분석에서는, 다양한 분류모델 전체를 돌려보기에는 한계가 있어 Pycaret이라는 Auto ML 사용하여 상위 모델 산출
  - 대개 자동으로 머신러닝 모델을 구축해주는 것을 Auto ML이라고 하나, Pycaret은 low code ML 라이브러리 측에 든다고 할 수 있음
  - \* 다양한 모델의 성능을 평가하고 그중에서 순위를 정해 최적화하는 과정을 적은 코드로 실현하게 해줌
  - \* n개의 머신러닝 모델을 자동으로 돌려 가장 효과적인 모델과 파라미터를 찾아 준다는 장점
  - classification(분류), regression(회귀), clustering(군집), anomaly detection(이상감지), NLP(자연어처리), association rules(연관규칙) 지원

- setup(): 불균형 데이터 조정, 이상치 삭제, 다중공선성 삭제, 군집화, PCA, 정규화, 인코딩 및 디코딩, Fold 수, 타겟, 데이터셋 등 총 62개의 파라미터 조정
- compare\_models(sort=''): acc, auc, recall, prec, f1, kappa 등의 성능 지표에 따라 가장 우수한 모델부터 내림차순 정렬

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
qda	Quadratic Discriminant Analysis	0.9056	0.9494	1.0000	0.8494	0.9163	0.8111	0.8298
et	Extra Trees Classifier	0.9108	0.9777	0.8778	0.9393	0.9064	0.8216	0.8248
rf	Random Forest Classifier	0.8886	0.9452	0.8556	0.9165	0.8839	0.7772	0.7803
gbc	Gradient Boosting Classifier	0.8492	0.9374	0.8222	0.8690	0.8435	0.6985	0.7014
lightgbm	Light Gradient Boosting Machine	0.8324	0.9126	0.8000	0.8576	0.8251	0.6649	0.6700
dt	Decision Tree Classifier	0.8208	0.8209	0.8333	0.8172	0.8233	0.6417	0.6448
ada	Ada Boost Classifier	0.8157	0.8491	0.8111	0.8192	0.8145	0.6314	0.6323
ridge	Ridge Classifier	0.7435	0.0000	0.6667	0.7945	0.7206	0.4872	0.4968
lda	Linear Discriminant Analysis	0.7435	0.8156	0.6667	0.7945	0.7206	0.4872	0.4968
lr	Logistic Regression	0.7325	0.7933	0.6889	0.7611	0.7171	0.4655	0.4741
knn	K Neighbors Classifier	0.7267	0.8260	0.6111	0.8083	0.6924	0.4537	0.4716
nb	Naive Bayes	0.7151	0.8040	0.5667	0.8228	0.6626	0.4315	0.4601
dummy	Dummy Classifier	0.4971	0.5000	0.8000	0.4000	0.5333	0.0000	0.0000
svm	SVM - Linear Kernel	0.5640	0.0000	0.4778	0.5473	0.4374	0.1222	0.1492

[그림 3-5] Pycaret을 이용한 분류모델 산출

- evaluate\_model(): 제공되는 모든 plot 확인 가능

\* auc, threshold, pr, 혼동행렬, feature selection, learning curve, 변수 중요도, lift chart, gain chart 등 20여 가지 plot 제공

Plot Type:	Hyperparameters	AUC	Confusion Matrix	Threshold	Precision Recall
	Prediction Error	Class Report	Feature Selection	Learning Curve	Manifold Learning
	Calibration Curve	Validation Curve	Dimensions	Feature Importance	Feature Importance
	Decision Boundary	Lift Chart	Gain Chart	Decision Tree	KS Statistic Plot

[그림 3-6] 제공 plot 종류

## 4. 분석 결과

### 4.1. 변수 선택

#### 4.1.1. 회귀분석

- 단계적 선택법을 활용하여 변수 선택 진행
  - 종속변수가 '단속지점개수'일 경우 회귀모형은 유의확률 2.2e-16으로 유의하나, 회귀계수 추정치가 유의하지 않은 변수가 다수 존재
  - 이 모형으로 단계적 선택법을 진행했을 때, 28개의 변수 중 14개의 변수가 선택
  - 회귀모형이 유의확률 2.2e-16으로 유의하고, 상수항을 포함하여 유의한 회귀계수가 12개 존재하며, 수정된 결정계수가 0.6083임
  - 회귀모형과 각 계수가 유의하나, '단속지점개수' 변수의 분포가 연속형보다 이산형에 가까워 제대로 적합 되지 않을 것으로 판단하여 분류모델에 적합하는 것으로 결정
  - 분석 결과

```
Call:
lm(formula = Y ~ X1 + X2 + X5 + X9 + X11 + X12 + X14 + X17 +
   X19 + X21 + X22 + X23 + X27 + X28, data = data_count)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.4400	-1.1196	-0.0949	0.9251	5.4054

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-5.362e-01	7.361e-01	-0.728	0.467717
X1	-6.001e-05	3.410e-05	-1.760	0.080845 .
X2	9.470e-05	4.487e-05	2.111	0.036772 *
X5	-3.146e-02	1.126e-02	-2.794	0.006019 **
X9	-1.092e-03	3.306e-04	-3.305	0.001238 **
X11	7.127e-03	2.256e-03	3.160	0.001979 **
X12	-1.471e-01	6.261e-02	-2.349	0.020369 *
X14	1.710e-01	5.653e-02	3.025	0.003011 **
X17	9.605e-02	2.392e-02	4.016	0.000101 ***
X19	-2.376e-03	6.891e-04	-3.448	0.000768 ***
X21	4.789e-05	1.738e-05	2.755	0.006743 **
X22	-6.920e-02	2.419e-02	-2.861	0.004944 **
X23	-2.403e-02	1.445e-02	-1.663	0.098884 .
X27	1.337e-02	2.097e-03	6.374	3.17e-09 ***
X28	1.740e-02	6.718e-03	2.590	0.010728 *
--				

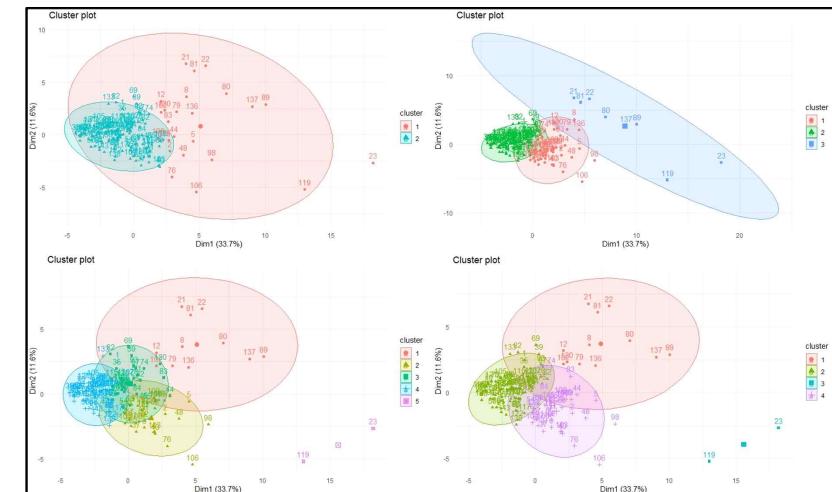
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.786 on 126 degrees of freedom  
 Multiple R-squared: 0.6474, Adjusted R-squared: 0.6083  
 F-statistic: 16.53 on 14 and 126 DF, p-value: < 2.2e-16

\* Y: 단속지점개수, X1: 주민등록인구, X2: 등록차량대수, X5: 베스정류장, X9: 주택, X11: 제2종생활근린시설, X12: 문화및집회시설, X14: 운수시설, X17: 업무시설, X19: 공장, X21: 교차로 교통량, X22: 상권\_관광여가오락, X23: 상권\_부동산, X27: 상권\_음식, X28: 상권\_학문교육

#### 4.1.2. 비 계층적 군집분석

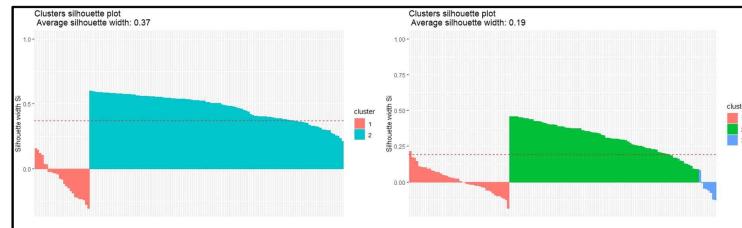
- 결합파일\_1095.csv를 이용하여 유clidean 거리 기준 군집분석 진행
  - \* Euclidean Distance: 가장 흔히 사용되는 거리 척도로 두 관측치 사이의 직선 최단 거리
- 수치형 컬럼은 컬럼별로 z-스코어 정규화 후 진행
  - \* z-스코어 정규화(z-score normalization): 기존 데이터를 평균은 0, 표준편차는 1인 형태의 데이터로 만드는 방법
- k-means를 이용하여 군집 2개부터 최대 9개까지 시도
- elbow method 상으로 4~5개의 군집이 적정하다고 판단
  - \* 가장 보편적으로 이용되는 방법으로 클러스터 내의 총변동을 설명하는 WCSS(Within Clusters Sum of Squares)를 이용



[그림 4-1] 비 계층적 군집분석 결과 시각화

- 군집 별 실루엣 계수가 모두 약 -0.4~0.5로 나옴

\* 실루엣 계수(Silhouette Coefficient) : 각 데이터 포인트와 주위 데이터 포인트들과의 거리 계산을 통해 값을 구하며, 군집 안에 있는 데이터들은 잘 모여있는지, 군집끼리는 서로 잘 구분되는지 클러스터링을 평가하는 척도로 활용



[그림 4-2] 비 계층적 군집분석 실루엣 계수 시작화

- 분석 시작화한 결과, 군집화가 제대로 이루어지지 않았고, 실루엣 계수가 작으므로 모델에 적합할 수 없음

#### 4.1.3. 계층적 군집분석

- 병합 군집(Agglomerative Clustering)

: 각각의 데이터 포인트를 하나의 클러스터로 지정하고, 지정된 개수의 클러스터가 남을 때까지 가장 비슷한 두 클러스터를 합쳐 나가는 알고리즘

- 군집화 전, 데이터의 특성을 반영하여 표준 정규화 실시

\* 표준 정규화(Robust Scaler): 평균, 분산 대신 중앙값과 사분위 값을 사용하여 이상치의 영향을 최소화하는 스케일링 기법

- linkage 함수에서 초기 데이터 배열을 전달하고, euclidean 지표를 매개변수로 사용

- 군집 수가 너무 많고, 앞서 진행한 비 계층적 군집과 같이 흩어져있어 사용 불 가능하다고 판단하여 기각

row_label_1	row_label_2	distance	no. of items in clust.
cluster 1	127.0	130.0	1.560474
cluster 2	126.0	141.0	1.711749
cluster 3	67.0	142.0	1.937906
cluster 4	28.0	64.0	1.949874
cluster 5	87.0	90.0	1.959209
...	...	...	...
cluster 136	252.0	275.0	28.681014
cluster 137	22.0	272.0	29.972919
cluster 138	276.0	277.0	37.345894
cluster 139	20.0	271.0	52.416187
cluster 140	278.0	279.0	94.337743

140 rows × 4 columns

[그림 4-3] 계층적 군집분석 결과

#### 4.1.4. 분류 모델링

- 단속지점 개수의 유무에 따라 0, 1로 인코딩한 [1095\_단속유무] 변수 생성

\* 단속지점 개수의 값이 0~17 사이로 분포되어 있으나, 연속적이라기보다는 이산적인 변수의 특성을 고려

- 모든 모델을 최적 파라미터로 돌리기엔 한계가 있어 Pycaret 사용

\* Pycaret: AutoML을 하게 해주는 파이썬 라이브러리, scikit-learn 패키지 기반으로 Classification, Regression, Clustering, Anomaly Detection 등 다양한 모델 지원

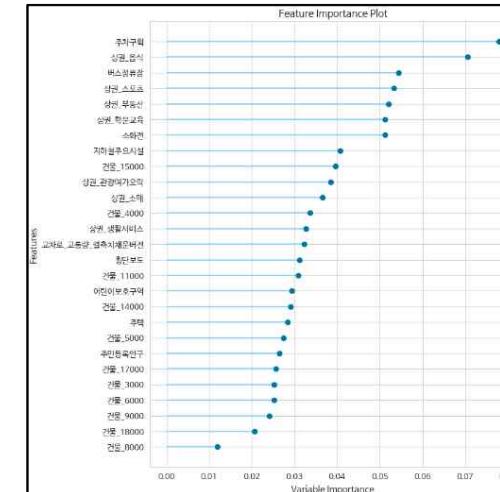
- setup에서 사용할 데이터, 타겟, 불균형조정(ADASYN, SMOTE), k-fold 횟수, 데이터 셋 분할 시 종속변수 비율 조정 등의 파라미터 조정

□ 불균형 데이터를 보완하기 위해 SMOTE, ADASYN 기법을 통해 오버 샘플링하여 균형을 맞춘 데이터 셋 2개를 만들어 성능 비교

- 모델 성능 지표로는 정밀도와 재현율의 조화 평균인 F1 Score 활용

\* 정확도보다 더 객관적으로 모델 성능을 평가할 수 있기 때문

- 변수중요도 및 confusion matrix, SHAP value 시작화



[그림 4-4] 변수중요도 Plot

변수 선택 결과

[표 4-1] 변수 선택 결과

구분	변수명	상세 설명
1	주민등록인구	지정 구역 내 주민등록인구
2	어린이보호구역	지정 구역 내 어린이보호구역 개수
3	주차구획	지정 구역 내 주차구획수
4	버스정류장	지정 구역 내 버스정류장 개수
5	지하철주요시설	지정 구역 내 지하철 주요시설 개수
6	소화전	지정 구역 내 소화전 개수
7	횡단보도	지정 구역 내 횡단보도 개수
8	주택	지정 구역 내 건물 용도 코드 1000~2003인 건물 개수
9	제1종근린생활시설	지정 구역 내 건물 용도 코드 3000인 건물 개수
10	제2종생활근린시설	지정 구역 내 건물 용도 코드 4000인 건물 개수
11	문화및집회시설	지정 구역 내 건물 용도 코드 5000인 건물 개수
12	종교시설	지정 구역 내 건물 용도 코드 6000인 건물 개수
13	병원	지정 구역 내 건물 용도 코드 9000인 건물 개수
14	업무시설	지정 구역 내 건물 용도 코드 14000인 건물 개수
15	숙박시설	지정 구역 내 건물 용도 코드 15000인 건물 개수
16	공장	지정 구역 내 건물 용도 코드 17000인 건물 개수
17	창고시설	지정 구역 내 건물 용도 코드 18000인 건물 개수
18	교차로 교통량	지정 구역 내 교차로 교통량
19	상권_관광여가오락	지정 구역 내 관광여가오락 상권 개수
20	상권_부동산	지정 구역 내 부동산 상권 개수
21	상권_생활서비스	지정 구역 내 생활서비스 상권 개수
22	상권_소매	지정 구역 내 소매 상권 개수
23	상권_스포츠	지정 구역 내 스포츠 상권 개수
24	상권_음식	지정 구역 내 음식 상권 개수
25	상권_학문교육	지정 구역 내 학문교육 상권 개수

## 4.2. 가중치 산정

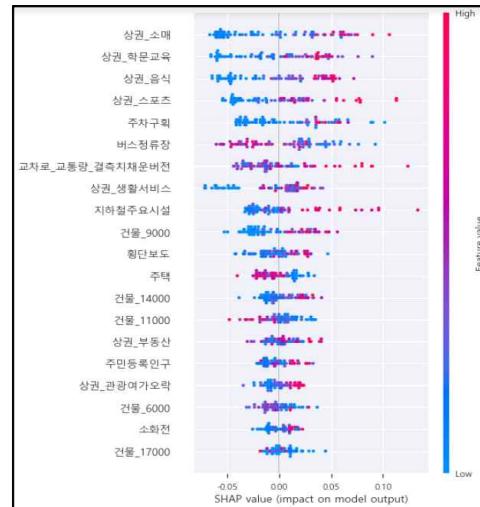
### 4.2.1. 가중치 산정 방법

- 가중치 산정 방법에는 주관적인 직접 할당법, 단순 비교법, 수리적 방법과 주관적 방법이 결합된 AHP와 회귀분석, 요인분석, 상관관계 분석을 통한 통계적 방법 등이 있음
- 본 연구에서는 보다 객관적인 판단을 하고, 주관을 최소화하기 위하여 통계적 방법 및 머신러닝을 활용한 방법을 통해 가중치 산정
- SHAP(Shapley Additive Explanations) values 활용
  - 게임이론을 바탕으로 하나의 특성에 대한 중요도를 알기 위해 여러 특성의 조합을 구성하고 해당 특성의 유무에 따른 평균적인 변화를 통해 얻어낸 값으로, 모든 가능한 조합에 대해서 하나의 특성의 기여도를 종합적으로 합한 값임
  - \* 한계기여도(marginal contribution)의 평균
  - \* 게임이론: 여러 주제가 서로 영향을 미치는 상황에서 서로가 어떤 의사결정이나 행동을 하는지에 대해 이론화한 것
  - 전체와 하나의 관측치를 비교하여 영향도를 구할 수 있고, 전체 데이터의 일부분 또는 하나의 인스턴스와 비교하여 영향도를 구할 수 있어 대조 설명이 가능함
  - $F$ 를 전체집합,  $S$ 를 전체집합에서  $i$ 번째 데이터가 빠진 나머지 집합,  $f_{S \cup \{i\}}(x_{S \cup \{i\}})$ 를  $i$ 번째 데이터를 포함한(=전체) 기여도,  $f_S(x_S)$ 를  $i$ 번째 데이터가 빠진 나머지 부분 집합의 기여도라고 할 때  $i$ 데이터에 대한 Shapley Value인  $\phi_i$ 는 다음과 같음
- \* 
$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)].$$
- 여러 모델을 돌려 나온 N개의 SHAP values 내의 영향도 고려
  - 원본 데이터의 종속변수 비율이 불균형하여 SMOTE, ADASYN 2가지 방식으로 오버 샘플링한 2개의 데이터 셋으로 학습
  - fold는 5와 10, 2가지 경우로 시행 및 학습
  - 샘플링 기법, fold 수에 따라 F1-Score 우선으로 타 성능지표도 우수한 상위 4개의 모델을 따로 산출(LGBM, Random Forest, Extra Trees, Logistic regression)

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TI (Sec)
qda	Quadratic Discriminant Analysis	0.9431	0.9702	1.0000	0.9110	0.9508	0.8848	0.8964
et	Extra Trees Classifier	0.9160	0.9812	0.8667	0.9635	0.9058	0.8319	0.8424
rf	Random Forest Classifier	0.9052	0.9645	0.9111	0.9117	0.9048	0.8103	0.8213
gbc	Gradient Boosting Classifier	0.8997	0.9739	0.8889	0.9135	0.8934	0.7992	0.8086
ada	Ada Boost Classifier	0.8824	0.9006	0.8888	0.8828	0.8824	0.7643	0.7702
lightgbm	Light Gradient Boosting Machine	0.8716	0.9552	0.8444	0.9042	0.8685	0.7434	0.7513
dt	Decision Tree Classifier	0.8209	0.8215	0.8000	0.8440	0.8172	0.6423	0.6487
knn	K Neighbors Classifier	0.7314	0.8137	0.6444	0.8121	0.7090	0.4635	0.4840
lr	Logistic Regression	0.7317	0.7636	0.6556	0.8139	0.6891	0.4619	0.4873
ridge	Ridge Classifier	0.6866	0.0000	0.5888	0.7631	0.6450	0.3735	0.3963
lda	Linear Discriminant Analysis	0.6866	0.7327	0.5889	0.7631	0.6450	0.3735	0.3963
dummy	Dummy Classifier	0.4971	0.5000	0.9000	0.4500	0.6000	0.0000	0.0000
nb	Naive Bayes	0.6542	0.7576	0.4778	0.7546	0.5669	0.3090	0.3373
svm	SVM - Linear Kernel	0.5708	0.0000	0.6556	0.6560	0.5637	0.1406	0.1714

[그림 4-5] SMOTE, fold 수 10일 때, 모델 매트릭스

- 샘플링 기법, fold 수, 모델에 따라 SHAP value의 편차가 조금씩 존재하여 이를 보완하기 위해 각 모델에서 나온 중요도 순위에 따라 상이한 점수를 부여하여 그 총합을 통해 최종 가중치 산정



[그림 4-6] 랜덤포레스트, SMOTE, fold 수 10일 때, SHAP values

- 독립변수가 종속변수에 영향을 미칠 때, 양/음의 방향인지 파악하고 양수와 음수 가중치 부여

#### 4.2.2. 산정 결과

- 상위 모델들의 SHAP values의 순위 산출 후, 각각 점수를 부여하고 그 총점에 따라 순위를 정해 가중치 부여

[표 4-2] SHAP values 기반 가중치 산정을 위한 점수 및 순위

변수명	점수	순위	최종 가중점수
상권_학문교육	48.1	1	4
상권_음식	45.15	2	4
주차구획	43	3	4
상권_스포츠	35.15	4	4
상권_소매	34.9	5	4
버스정류장(-)	31.05	6	4
소화전	30.7	7	4
상권_부동산	28.35	8	3
건물_14000	28	9	3
건물_17000(-)	23.7	10	3
상권_생활서비스	23.5	11	3
주택(-)	23.2	12	3
건물_4000(-)	19.6	13	3
건물_15000	18.9	14	3
건물_11000(-)	16.65	15	2
건물_9000	14.15	16	2
지하철주요시설	13.85	17	2
교차로_교통량	12.85	18	2
건물_3000	8.2	19	2
어린이보호구역(-)	6.3	20	2
횡단보도(-)	5.9	21	2
건물_18000	4.1	22	1
건물_5000	2.3	24	1
주민등록인구	1.05	25	1
건물_6000	1.05	26	1
상권_관광여가오락	0.3	27	1
건물_8000	0	28	0

\*(-)는 음의 가중치 부여

- 그 결과, -4에서 4까지 음의 가중치와 양의 가중치 부여

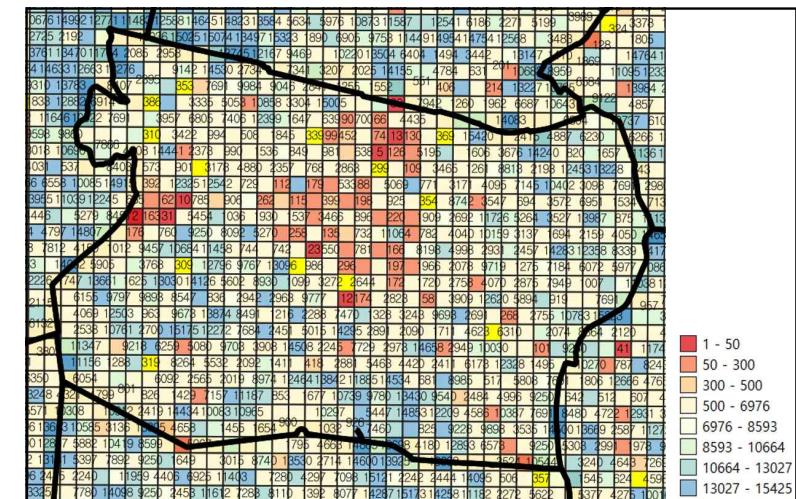
[표 4-3] SHAP values 기반 가중치 산정 결과

변수명	가중치
상권_소매, 상권_스포츠, 상권_음식, 상권_학문교육, 소화전, 주차구획	4
건물_14000(업무시설), 건물_15000(숙박시설), 상권_부동산, 상권_생활서비스	3
건물_3000(제1종근린생활시설), 건물_9000(병원), 교차로_교통량, 지하철 주요시설	2
건물_5000(문화및집회시설), 건물_6000(종교시설), 건물_18000(창고시설), 상권_관광여가오락, 주민등록인구	1
어린이보호구역, 횡단보도	-2
건물_4000(제2종근린생활시설), 건물_17000(공장), 주택	-3
버스정류장	-4

### 4.3. 최적 경로 선정

#### 4.3.1. 총점 계산

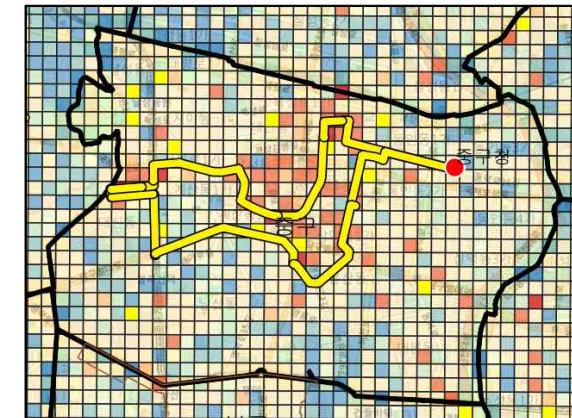
- Q-GIS를 이용하여 데이터 결합
  - 포인트 레이어 => [폴리곤에 포함하는 포인트 개수 계산] 이용하여 격자 내 점 개수 산출
  - 직전 단계에서 산출된 개수 레이어를 폴리곤으로 지정하여 다음 포인트 레이어 점 개수 산출
  - 전처리데이터.csv로 내보내기
- Python을 이용하여 총점 계산
  - 모든 변수들의 데이터 셋에 '가중점수' 컬럼 추가하여 가중치 입력
  - 점수 컬럼 Min-Max 정규화
  - 가중치 적용하여 총점 및 순위 계산
- Q-GIS로 결과 시각화
  - 총점에 따라 [단계구분 시각화]를 이용하여 단계별로 다른 색상으로 격자 시각화
  - 총점이 높은 구역(불법주정차 다발 구역)은 붉은색으로 표시
  - 각 격자마다 순위를 확인할 수 있도록 라벨링 및 단계별 심볼 설정



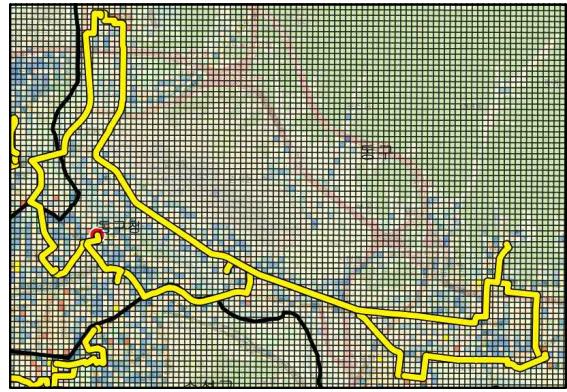
[그림 4-7] 대구시 중구 순위 격자 시각화 결과 (전체 시각화 결과는 부록 첨부)

#### 4.3.2. 최적 경로 도출

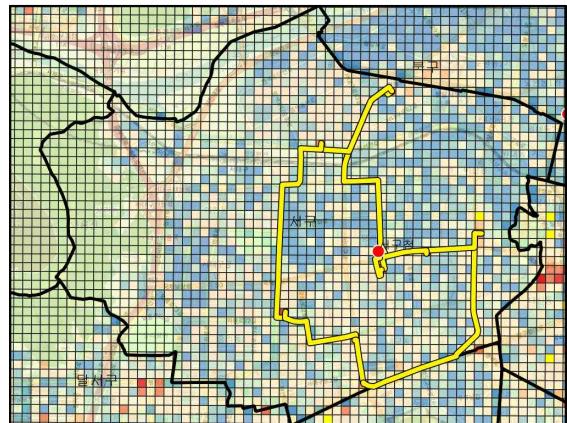
- QNEAT3 플러그인 - [shortest path(point to point)] 도구 활용하여 구청에서 시작하여 불법주정차 다발 구역을 지나는 최단거리 최적 경로 산출
- 네트워크 레이어를 '대구 도로망'으로 설정, 시작과 마지막 지점은 지나야 할 격자로 설정하여 도로망 기준으로 최적 경로 도출
- 구별 최적 경로 결과



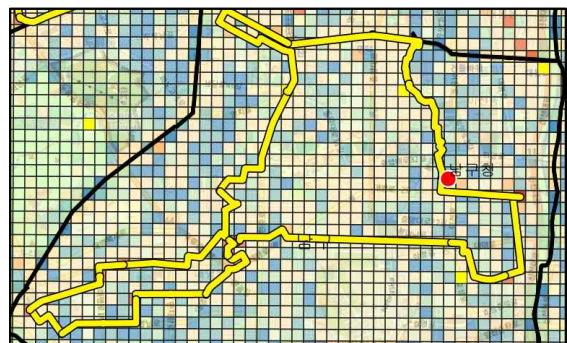
[그림 4-8] 대구시 중구 불법주정차 다발 구역 최적 단속경로



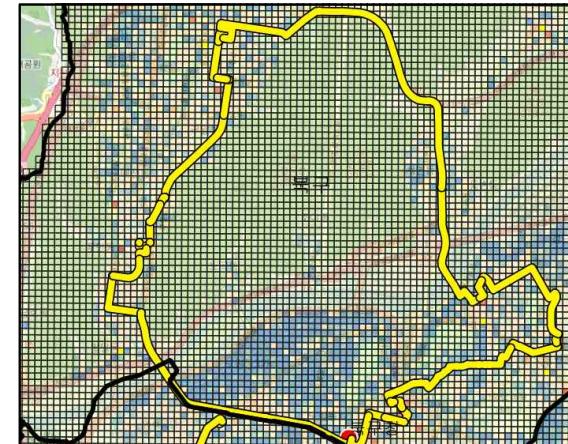
[그림 4-9] 대구시 동구 불법주정차 다발 구역 최적 단속경로



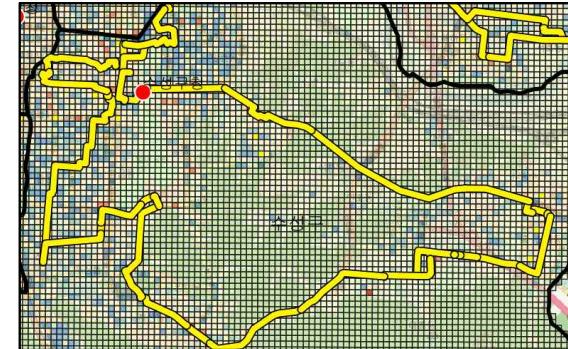
[그림 4-10] 대구시 서구 불법주정차 다발 구역 최적 단속경로



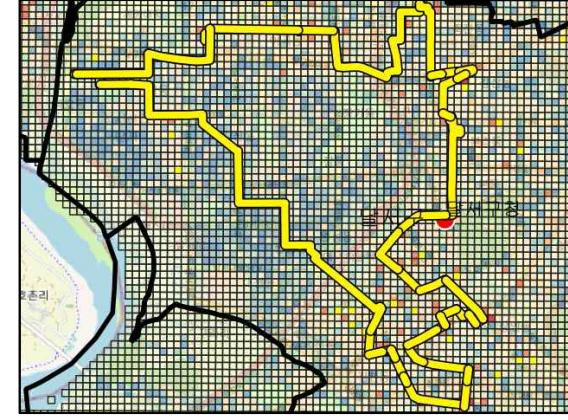
[그림 4-11] 대구시 남구 불법주정차 다발 구역 최적 단속경로



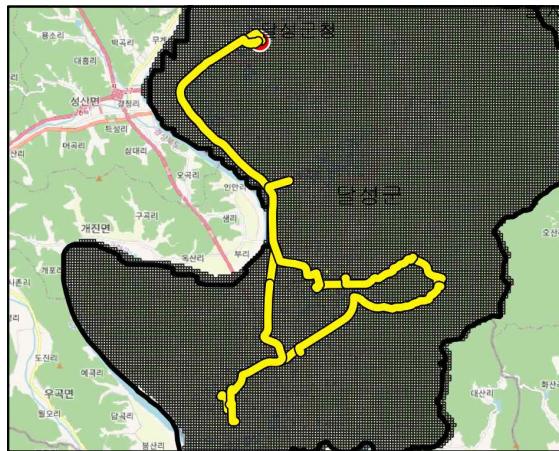
[그림 4-12] 대구시 북구 불법주정차 다발 구역 최적 단속경로



[그림 4-13] 대구시 수성구 불법주정차 다발 구역 최적 단속경로



[그림 4-14] 대구시 달서구 불법주정차 다발 구역 최적 단속경로



[그림 4-15] 대구시 달성군 불법주정차 다발 구역 최적 단속경로



[그림 4-16] 대구시 달성군 불법주정차 다발 구역 최적 단속경로

## 5. 활용 방안

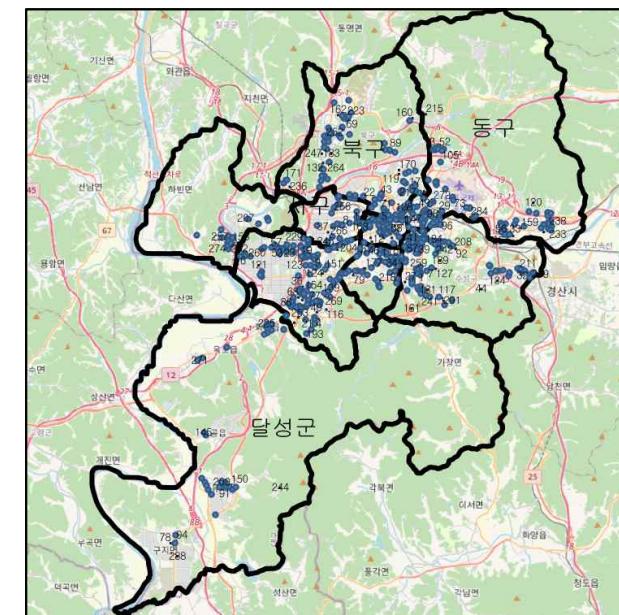
### 5.1. 문제점 개선 방안

#### 5.1.1. 불법주정차 단속체계 정비의 가이드라인 제공

- 최적 단속경로 모델 제안으로 각 구청 교통과마다 존재했던 단속 기준에 대한 편차를 줄이는 것에 기여
- 기존에는 불법주정차 단속 차량 경로의 일정한 기준 부재, 민원인의 신고를 근거로 한 단속 조치의 한계점이 있었으나 가이드라인 제공으로 극복 가능
- 각 구청에서 출발하여 단속 필수구역을 거쳐 다시 돌아오는 최적 경로를 도출
- 적은 인력으로 효율적인 불법주정차 관련 업무 수행 가능

#### 5.1.2. CCTV 추가 설치 필요

- 더 많은 지역에 탑재 및 고정형 CCTV 필요
  - 지역마다 CCTV 수의 편차가 큼



[그림 5-1] 대구시 고정형 CCTV 설치 현황

- 누적 단속 수가 많은 지역임에도 불구하고 CCTV의 수가 적음



[그림 5-2] 불법주정차 다발 구역 부근 CCTV 설치 현황

- 주차구획의 증대만이 불법주정차 완화 요인이 될 수 없음에도 불구하고 주차구획 증대에만 치중된 현 정책 보완 필요

## 5.2. 업무 활용 방안

### 5.2.1. 지방자치단체 업무 활용

- 작성한 주정차 단속경로를 바탕으로 민원이 많은 시간 이전에 미리 단속하여 단속 업무 부담을 줄임
- 분석을 바탕으로 고정식 CCTV에 대한 추가 및 이동 설치 고려, 버스 탑재형 CCTV를 추가 설치할 버스 노선 파악 가능

### 5.2.2. 표준 모델 활용

- 대구광역시를 대상으로 최적 경로 도출 모델을 만들었으며, 데이터 형식에 맞추어 다른 지역의 데이터를 수집하여 활용 가능

## 6. 참고자료(Reference)

도로교통법

도로교통법

[시행 2022. 7. 12.] [법률 제18741호, 2022. 1. 11., 일부개정]

경찰청 (교통기획계(법제총괄)) 02-3150-2251

경찰청 (교통운영계(신호, 안전고지, 도로공사신고)) 02-3150-2753

경찰청 (운전 면허계(운전면허)) 02-3150-2253

경찰청 (도통안전계(안전, 단속, 어린이통학버스)) 02-3150-2252

경찰청 (교통조사계(교통사고조사)) 02-3150-2552

**제32조(정차 및 주차의 금지)** 모든 차의 운전자는 다음 각 호의 어느 하나에 해당하는 곳에서는 차를 정차하거나 주차하여서는 아니 된다. 다만, 이 법이나 이 법에 따른 명령 또는 경찰공무원의 시지를 따른 경우와 위험방지를 위하여 일시정지하는 경우에는 그러하지 아니하다. <개정 2018. 2. 9., 2020. 10. 20., 2020. 12. 22. >

1. 교차로 · 횡단보도 · 건널목이나 보도와 차도가 구분된 도로의 보도(「주차장법」에 따라 차도와 보도에 걸쳐서 설치된 노상주차장은 제외한다)

2. 교차로의 가상자리나 도로의 모퉁이로부터 5미터 이내인 곳

3. 안전지대가 설치된 도로에서는 그 안전지대의 사방으로부터 각각 10미터 이내인 곳

4. 버스여객자동차의 정류지(停留地)임을 표시하는 기둥이나 표지판 또는 선이 설치된 곳으로부터 10미터 이내인 곳. 다만, 버스여객자동차의 운전자가 그 버스여객자동차의 운행시간 중에 운행노선에 따르는 정류장에서 승객을 태우거나 내리기 위하여 차를 정차하거나 주차하는 경우에는 그러하지 아니하다.

5. 건널목의 가상자리 또는 횡단보도로부터 10미터 이내인 곳

6. 다음 각 목의 곳으로부터 5미터 이내인 곳  
가. 「소방기본법」 제10조에 따른 소방용수시설 또는 비상소화장치가 설치된 곳  
나. 「화재예방, 소방시설 설치·유지 및 안전관리에 관한 법률」 제2조제1항제1호에 따른 소방시설로서 대통령령으로 정하는 시설이 설치된 곳

7. 시·도경찰청이 도로에서의 위험을 방지하고 교통의 안전과 원활한 소통을 확보하기 위하여 필요하다고 인정하여 지정한 곳

8. 시장등이 제12조제1항에 따라 지정한 이단이 호보구역

[전문개정 2011. 6. 8.]

**제32조(정차 및 주차의 금지)** 모든 차의 운전자는 다음 각 호의 어느 하나에 해당하는 곳에서는 차를 정차하거나 주차하여서는 아니 된다. 다만, 이 법이나 이 법에 따른 명령 또는 경찰공무원의 시지를 따른 경우와 위험방지를 위하여 일시정지하는 경우에는 그러하지 아니하다. <개정 2018. 2. 9., 2020. 10. 20., 2020. 12. 22., 2021. 11. 30. >

1. 교차로 · 횡단보도 · 건널목이나 보도와 차도가 구분된 도로의 보도(「주차장법」에 따라 차도와 보도에 걸쳐서 설치된 노상주차장은 제외한다)

2. 교차로의 가상자리나 도로의 모퉁이로부터 5미터 이내인 곳

3. 안전지대가 설치된 도로에서는 그 안전지대의 사방으로부터 각각 10미터 이내인 곳

4. 버스여객자동차의 정류지(停留地)임을 표시하는 기둥이나 표지판 또는 선이 설치된 곳으로부터 10미터 이내인 곳. 다만, 버스여객자동차의 운전자가 그 버스여객자동차의 운행시간 중에 운행노선에 따르는 정류장에서 승객을 태우거나 내리기 위하여 차를 정차하거나 주차하는 경우에는 그러하지 아니하다.

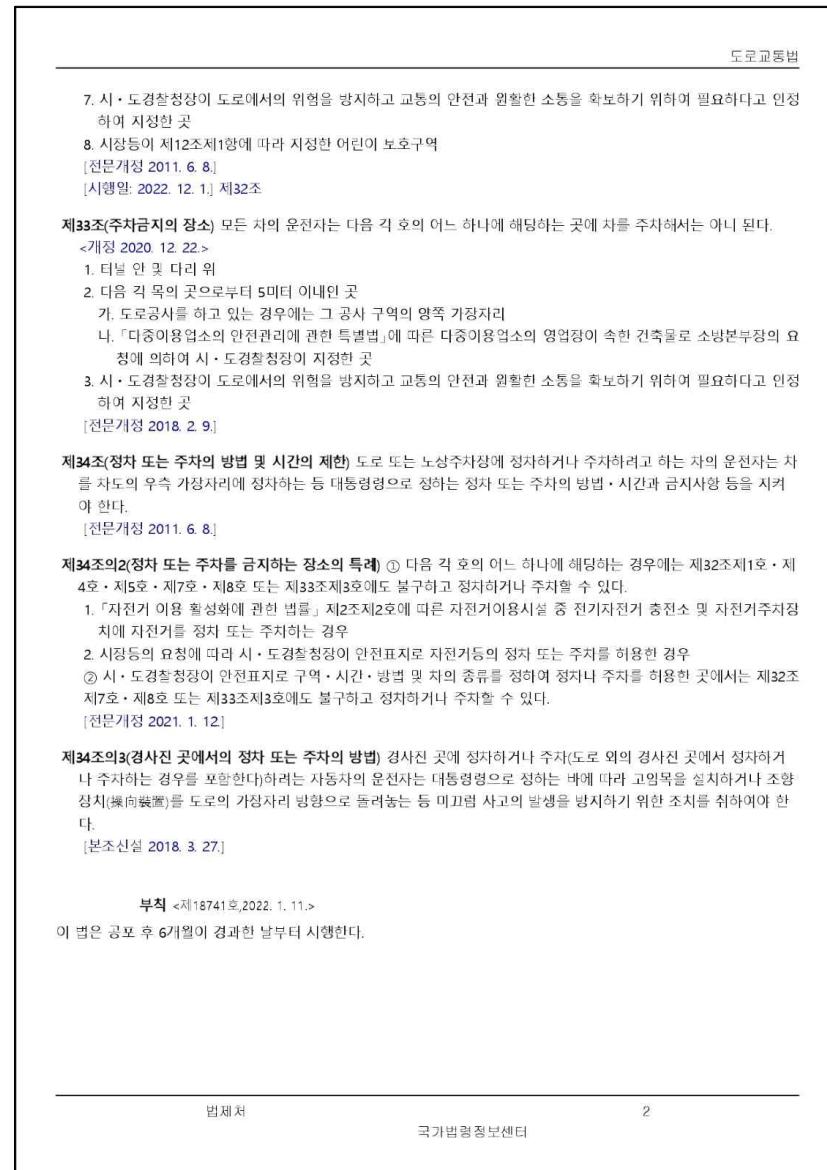
5. 건널목의 가상자리 또는 횡단보도로부터 10미터 이내인 곳

6. 다음 각 목의 곳으로부터 5미터 이내인 곳  
가. 「소방기본법」 제10조에 따른 소방용수시설 또는 비상소화장치가 설치된 곳  
나. 「소방시설 설치 및 관리에 관한 법률」 제2조제1항제1호에 따른 소방시설로서 대통령령으로 정하는 시설이 설치된 곳

법제처

국가법령정보센터

[그림 6-1] 도로교통법 제32조



[그림 6-2] 도로교통법 제32조, 제33조, 제34조

**주정차위반 단속**

- 관계법령 : 도로교통법 제32조, 제33조, 제34조의 주정차 금지 위반

**단속대상**

- 주차와 정차가 금지되는 장소
  - 주정차금지표지판 설치 지역 또는 황색실선, 황색이중실선이 표시된 곳
  - 교차로, 횡단보도, 건널목, 보도와 차도가 구분된 도로의 보도
  - 교차로의 가장자리나 도로의 모퉁이로부터 50미터 이내의 곳
  - 도로안전지대, 버스정류장, 건널목, 횡단보도의 10미터 이내의 곳
  - 소방시설(소방용수시설, 비상소화장치 등)으로부터 50미터 이내의 곳
- 주차가 금지되는 장소
  - 주차금지표지판 설치 지역 또는 황색 점선이 표시된 곳
  - 터널 안 및 다리 위

**과태료 부과금액**

차종별	과태료금액			
	일반지역	어린이 보호구역	노인 보호구역	소화전
승용자동차, 4륜이하 화물자동차	40,000원	120,000원	80,000원	80,000원
승합자동차, 4륜초과화물자동차, 특수자동차, 건설기계	50,000원	130,000원	90,000원	90,000원

\* 질서위반행위규제법에 따라 의견제출기간내 과태료를 자진납부 할 때에는 20% 경감받을 수 있습니다.

\* 같은 장소에서 2시간 이상 불법주정차를 하는 경우 1만원 가중 부과됩니다.

[그림 6-3] 주정차 위반 단속



[그림 6-4] 불법주정차 단속 흐름도



[그림 6-5] 단속 권한 및 단속체계



[그림 6-6] 도로주정차선별 주정차 가능 여부

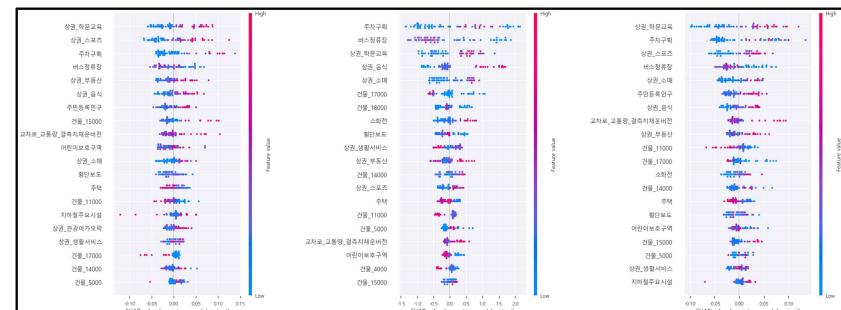
## 7. 부록

### 7.1. 추가 분석 결과

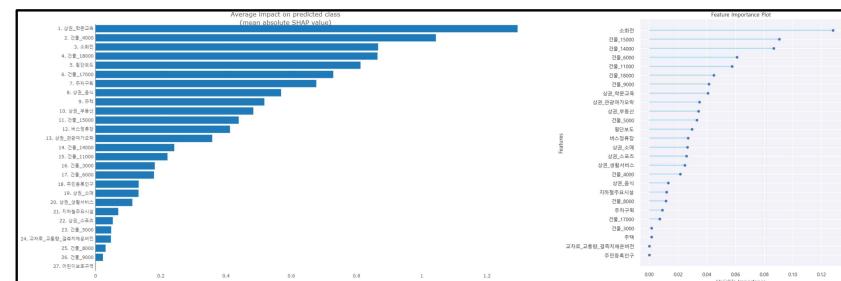
#### 7.1.1. 가중치 산정에 이용한 모델 평가 및 해석

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)	
qda	Quadratic Discriminant Analysis	0.8475	0.8933	1.0000	0.7729	0.8704	0.6946	0.7312	0.0060
et	Extra Trees Classifier	0.7944	0.8950	0.7242	0.8625	0.7792	0.5894	0.6076	0.0560
rf	Random Forest Classifier	0.7813	0.8607	0.7233	0.8321	0.7682	0.5628	0.5747	0.0740
gbc	Gradient Boosting Classifier	0.7553	0.8616	0.6983	0.7908	0.7408	0.5111	0.5155	0.0260
dt	Decision Tree Classifier	0.7411	0.7413	0.7358	0.7458	0.7375	0.4822	0.4862	0.0080
lightgbm	Light Gradient Boosting Machine	0.7284	0.7068	0.7108	0.7623	0.7284	0.4571	0.4668	0.0100
ada	Ada Boost Classifier	0.7088	0.7848	0.6325	0.7472	0.6833	0.4186	0.4246	0.0260
lr	Logistic Regression	0.6619	0.6601	0.5908	0.6917	0.6304	0.3240	0.3299	0.0540
knn	K Neighbors Classifier	0.6619	0.7572	0.5533	0.7431	0.6193	0.3254	0.3497	0.0100
nb	Naive Bayes	0.6497	0.7337	0.5033	0.7330	0.5806	0.3029	0.3267	0.0060
ridge	Ridge Classifier	0.6097	0.0000	0.4875	0.6666	0.5565	0.2207	0.2342	0.0040
lda	Linear Discriminant Analysis	0.6097	0.6107	0.4875	0.6666	0.5565	0.2207	0.2342	0.0100
dummy	Dummy Classifier	0.4968	0.5000	0.8000	0.4000	0.5333	0.0000	0.0000	0.0040
svm	SVM - Linear Kernel	0.5368	0.0000	0.6533	0.4329	0.4974	0.0800	0.0934	0.0060

[그림 7-1] ADASYN, fold 5일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스



[그림 7-2] ADASYN, fold 5일 때 SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)



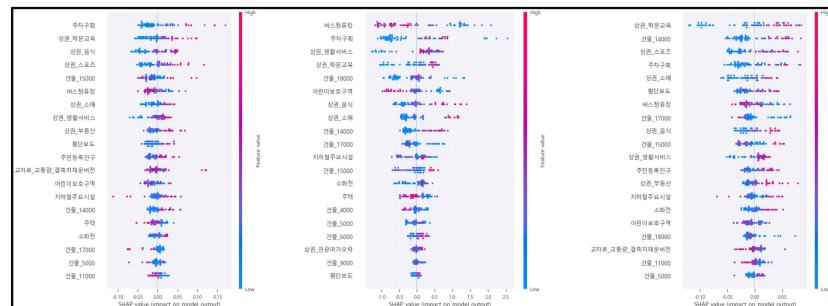
[그림 7-3] ADASYN, fold 5일 때, SHAP Values 및 변수중요도(Logistic Regression)

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)	
qda	Quadratic Discriminant Analysis	0.9000	0.9134	1.0000	0.8419	0.9115	0.7990	0.8205	0.0060
et	Extra Trees Classifier	0.8417	0.9368	0.7875	0.8990	0.8229	0.6801	0.7021	0.0630
lightgbm	Light Gradient Boosting Machine	0.8354	0.8835	0.7768	0.8907	0.8120	0.6696	0.6914	0.0110
rf	Random Forest Classifier	0.8150	0.8815	0.7875	0.8431	0.8050	0.6278	0.6418	0.0720
ada	Ada Boost Classifier	0.7742	0.8290	0.7643	0.8050	0.7712	0.5479	0.5684	0.0250
gbc	Gradient Boosting Classifier	0.7892	0.9049	0.7357	0.8235	0.7667	0.5772	0.5906	0.0360
dt	Decision Tree Classifier	0.7275	0.7321	0.6214	0.8226	0.6917	0.4588	0.4872	0.0050
knn	K Neighbors Classifier	0.6888	0.7759	0.6554	0.7230	0.6752	0.3737	0.3866	0.0090
lr	Logistic Regression	0.6425	0.6152	0.5807	0.7097	0.6042	0.2901	0.3079	0.0540
nb	Naive Bayes	0.6692	0.7315	0.5321	0.7577	0.6024	0.3420	0.3691	0.0050
ridge	Ridge Classifier	0.6158	0.0000	0.5464	0.6414	0.5834	0.2352	0.2377	0.0060
lda	Linear Discriminant Analysis	0.6158	0.6116	0.5464	0.6414	0.5834	0.2352	0.2377	0.0050
svm	SVM - Linear Kernel	0.5554	0.0000	0.6696	0.5109	0.5487	0.1123	0.1142	0.0060
dummy	Dummy Classifier	0.4700	0.5000	0.5000	0.2367	0.3212	0.0000	0.0000	0.0070

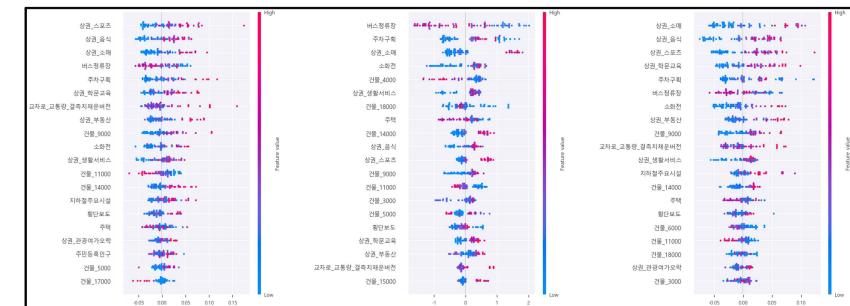
[그림 7-4] ADASYN, fold 10일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)	
qda	Quadratic Discriminant Analysis	0.9383	0.9657	1.0000	0.8967	0.9441	0.8762	0.8858	0.0100
et	Extra Trees Classifier	0.9329	0.9863	0.8778	0.9889	0.9286	0.8659	0.8733	0.0520
gbc	Gradient Boosting Classifier	0.9051	0.9504	0.8889	0.9221	0.9037	0.8103	0.8133	0.0280
rf	Random Forest Classifier	0.8989	0.9645	0.8778	0.9176	0.8968	0.7978	0.7992	0.0660
ada	Ada Boost Classifier	0.8659	0.8991	0.8444	0.8898	0.8615	0.7322	0.7401	0.0300
dt	Decision Tree Classifier	0.8494	0.8500	0.8556	0.8570	0.8506	0.6990	0.7095	0.0060
lightgbm	Light Gradient Boosting Machine	0.8487	0.9264	0.8444	0.8577	0.8489	0.6974	0.7012	0.2300
knn	K Neighbors Classifier	0.7202	0.8019	0.7000	0.7428	0.7156	0.4390	0.4451	0.4700
lr	Logistic Regression	0.6867	0.7451	0.6000	0.7318	0.6578	0.3742	0.3818	1.1260
nb	Naive Bayes	0.6651	0.7196	0.5000	0.7539	0.5962	0.3315	0.3548	0.0060
ridge	Ridge Classifier	0.6246	0.0000	0.5222	0.6717	0.5858	0.2498	0.2589	0.0060
lda	Linear Discriminant Analysis	0.6246	0.6994	0.5222	0.6717	0.5858	0.2498	0.2589	0.0080
dummy	Dummy Classifier	0.4971	0.5000	0.8000	0.4000	0.5333	0.0000	0.0000	0.0060
svm	SVM - Linear Kernel	0.5533	0.0000	0.4556	0.6133	0.3893	0.1033	0.1719	0.0060

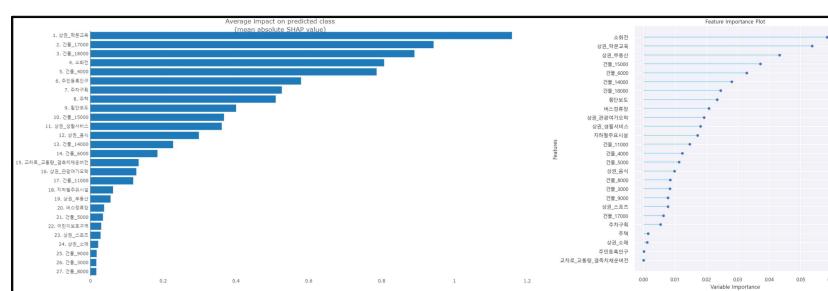
[그림 7-7] SMOTE, fold 5일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스



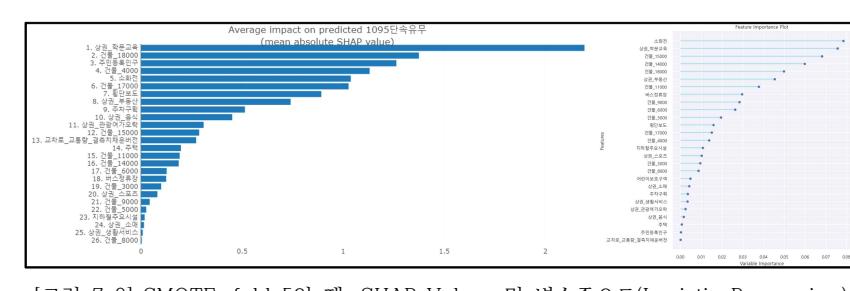
[그림 7-5] ADASYN, fold 10일 때 SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)



[그림 7-8] SMOTE, fold 5일 때, SHAP Values (왼쪽부터 Extra Trees, LGBM, Random Forest)



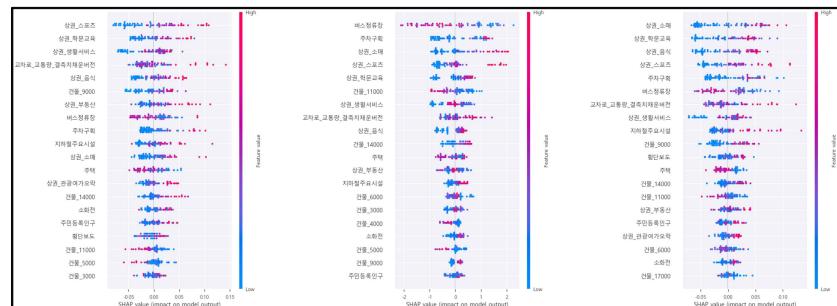
[그림 7-6] ADASYN, fold 10일 때, SHAP Values 및 변수중요도(Logistic Regression)



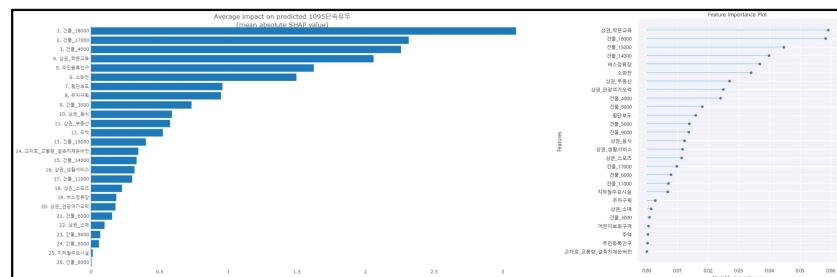
[그림 7-9] SMOTE, fold 5일 때, SHAP Values 및 변수중요도(Logistic Regression)

Model		Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
qda	Quadratic Discriminant Analysis	0.9431	0.9702	1.0000	0.9110	0.9508	0.8848	0.8964	0.0040
et	Extra Trees Classifier	0.9160	0.9812	0.8667	0.9635	0.9058	0.8319	0.8424	0.0640
rf	Random Forest Classifier	0.9052	0.9645	0.9111	0.9117	0.9048	0.8103	0.8213	0.0680
gbc	Gradient Boosting Classifier	0.8997	0.9739	0.8889	0.9135	0.8934	0.7992	0.8086	0.0420
ada	Ada Boost Classifier	0.8824	0.9006	0.8889	0.8828	0.8824	0.7643	0.7702	0.0360
lightgbm	Light Gradient Boosting Machine	0.8716	0.9552	0.8444	0.9042	0.8685	0.7434	0.7513	0.0110
dt	Decision Tree Classifier	0.8209	0.8215	0.8000	0.8440	0.8172	0.6423	0.6487	0.0050
knn	K Neighbors Classifier	0.7314	0.8137	0.6444	0.8121	0.7090	0.4635	0.4840	0.0100
lr	Logistic Regression	0.7317	0.7636	0.6556	0.8139	0.6891	0.4619	0.4873	0.0650
ridge	Ridge Classifier	0.6866	0.0000	0.5889	0.7631	0.6450	0.3735	0.3963	0.0050
lda	Linear Discriminant Analysis	0.6866	0.7327	0.5889	0.7631	0.6450	0.3735	0.3963	0.0060
dummy	Dummy Classifier	0.4971	0.5000	0.9000	0.4500	0.6000	0.0000	0.0000	0.0050
nb	Naive Bayes	0.6542	0.7576	0.4778	0.7546	0.5669	0.3090	0.3373	0.0050
svm	SVM - Linear Kernel	0.5706	0.0000	0.6556	0.6560	0.5637	0.1406	0.1714	0.0040

[그림 7-10] SMOTE, fold 10일 때, pycaret이 돌린 모든 모델들의 성능 매트릭스

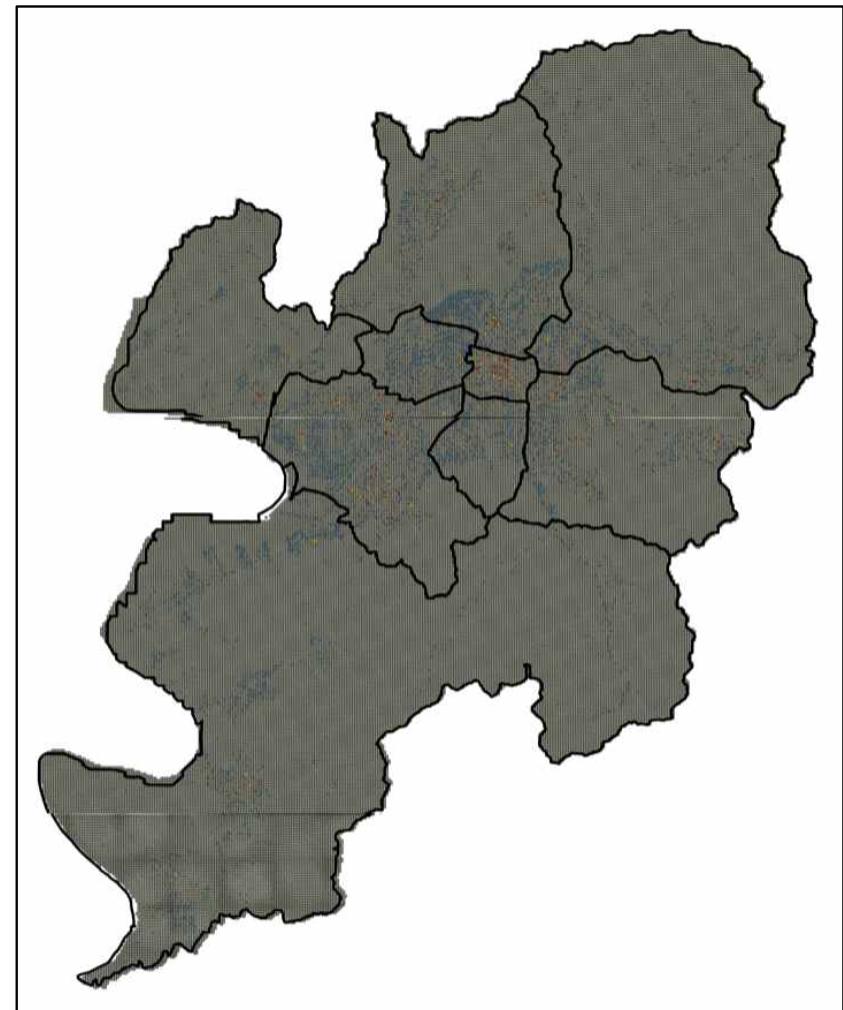


[그림 7-11] SMOTE, fold 10일 때 SHAP Values (왼쪽부터) Extra Trees, LGBM, Random Forest)

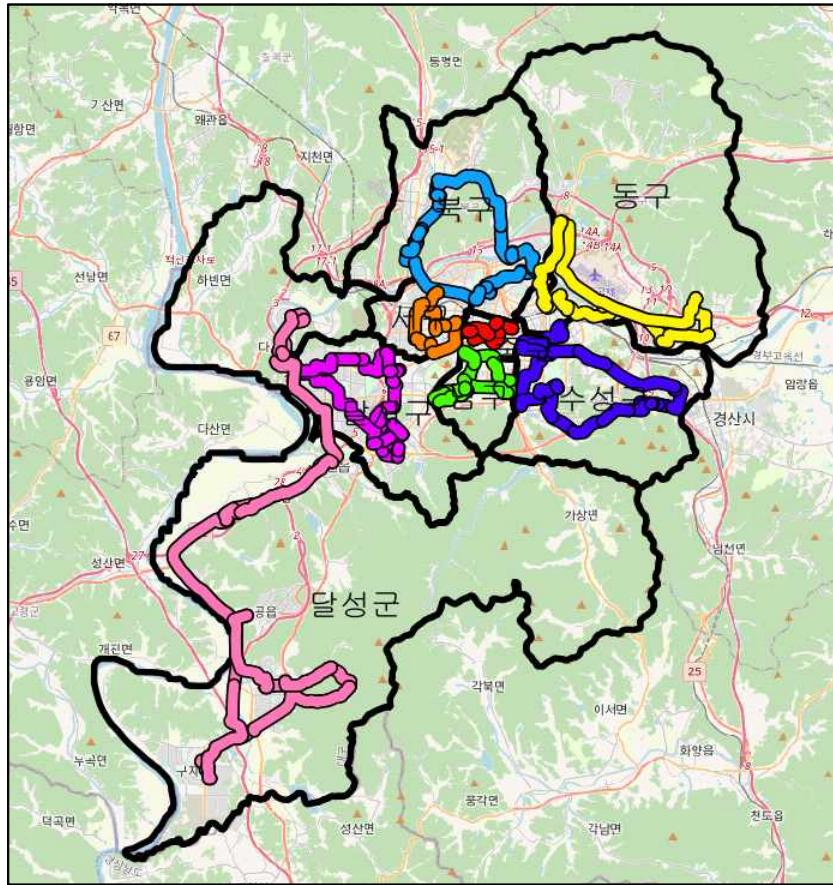


[그림 7-12] SMOTE, fold 10일 때, SHAP Values 및 변수중요도(Logistic Regression)

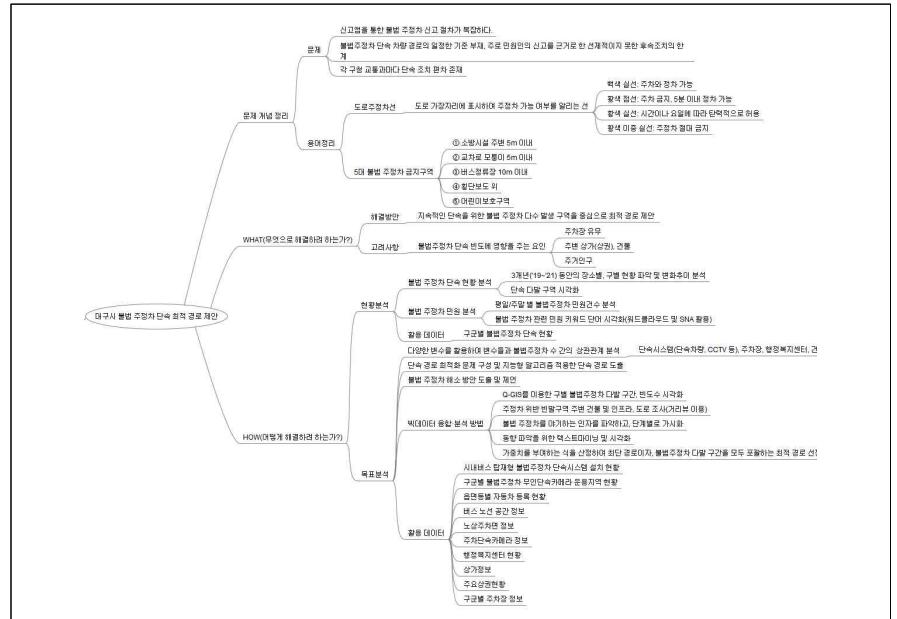
#### 7.1.2. 최적 경로 선정



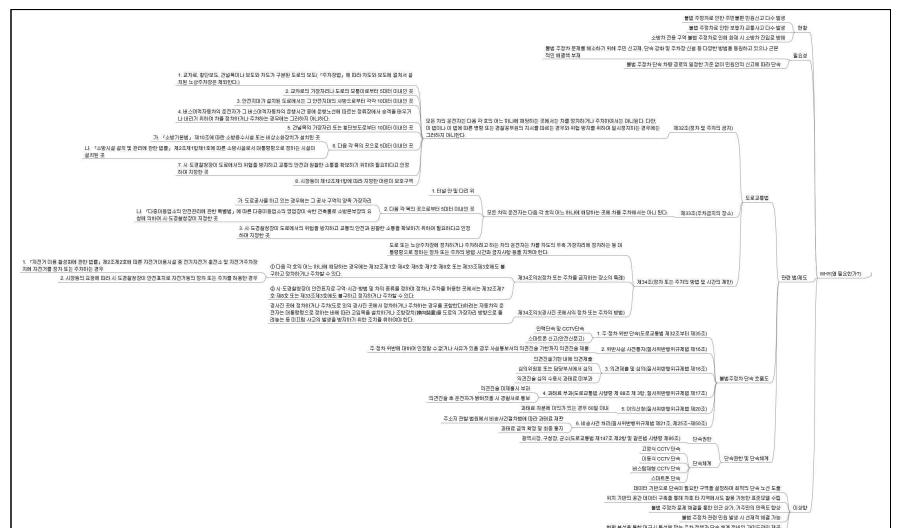
[그림 7-13] 대구시 격자 순위 시각화 결과



## 7.2. 주제설계를 위한 마인드맵



[그림 7-15] 분석논리모형 마인드맵



[그림 7-16] 분석논리모형 마인드맵

### 7.3. 분석 상세코드

#### 7.3.1. 단속 현황 전처리(2.3.1)

```
# 구별 데이터 불러오기
df = pd.read_csv('데이터셋.csv', encoding = 'cp949')
df

# 위반일자 칼럼 데이터타입 문자열에서 datetime으로 변환
df['위반일자'] = pd.to_datetime(df['위반일자'], format='%Y-%m-%d')
df['위반일자']

# 연도만 뽑고, 2019 이전 데이터는 삭제
df['Year'] = df['위반일자'].dt.year
df = df.drop(df[df['Year'] < 2019].index)
df['Year'].value_counts()

# 장소별 누적 수 출력
num = df['위반장소명'].value_counts()
num= num.to_frame()
num.reset_index(drop = False, inplace = True)
num.columns = ['위반장소명','누적수']

# 누적수 1095 이하 삭제 후 csv 내보내기
df = num.drop(num[num['누적수'] < 1095].index)
df.to_csv("n구누적.csv", encoding = 'cp949')
```

#### 7.3.2. 좌표변환(2.3.1)

```
# 파일 불러오기
df = pd.read_csv('/content/좌표변환할 데이터셋.csv', encoding = 'cp949')
# 주소만 address에 할당
address = df['주소']
print(address.head())
# 한국으로 좌표 설정
geo_local = Nominatim(user_agent='South Korea')
# 위도, 경도 반환하는 함수 생성
def geocoding(address):
    try:
        geo = geo_local.geocode(address)
        x_y = [geo.latitude, geo.longitude]
    except:
        x_y = [0,0]
    return x_y
```

#### # 위경도 빈 리스트 생성

```
latitude = []
longitude = []

# 좌표를 리스트에 추가하는 반복문
for i in address:
    latitude.append(geocoding(i)[0])
    longitude.append(geocoding(i)[1])

# 리스트 df 변환 및 기존 df에 병합
la = pd.DataFrame(latitude)
lo = pd.DataFrame(longitude)
df = pd.concat([df, la, lo], axis = 1)
```

#### 7.3.3. 교통량 결측치(2.3.7)

```
# 파일 불러오기
df = pd.read_csv('/content/교통량_결측치.csv', encoding = 'cp949')
# 결측값 확인
df.isnull().sum()
# 결측치 채우기 위해서 행정동명 라벨인코딩
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(df['행정동명'])
le_encoded = le.transform(df['행정동명'])
new_cat = pd.DataFrame(le_encoded, columns = ['행정동명_인코딩'])
le.classes_ # array형태를 df로 변환
# 기존 데이터 셋이랑 결합
df = pd.concat([df, new_cat], axis = 1)
df = df.drop(['행정동명', '건물'], axis=1) # 건물은 지금 처리할 부분이 아니니까 드롭, 행정동명은 인코딩 했으니까 드롭

# mice 방법을 사용하기 위해 모듈 임포트(기존에는 mice 모듈이 있었는데 최근에 업데이트
# 하며 사라져서 직접 모듈 임포트해서 만드는 형식으로 합니다)
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

lr = LinearRegression() # 선형회귀모델 할당
imp = IterativeImputer(estimator = lr, verbose = 2, max_iter = 30, tol = 1e-10,
imputation_order='roman')
```

```
# df%/% 피팅
imp.fit(df)
# a에 할당해서 피팅한 값 저장 및 변환
a = imp.transform(df)
a = pd.DataFrame(a)
# csv로 내보내서 엑셀에서 7번 칼럼 복사 붙여넣기하여 수정
a.to_csv('교통량결측치채움.csv', encoding = 'cp949')
```

#### 7.3.4. 다중회귀분석 단계적 선택법(4.1.1)

```
#종속변수: 단속지점개수
data_count = read.csv("C:/Users/user/Desktop/data_point_count.csv", header=TRUE,
sep=",", fileEncoding = "euc-kr")
data_count
# 전체 변수 모델 적합
lm_out_count<- lm(Y~, data = data_count)
lm_out_count
# 회귀모형 적합도, 회귀계수 유의성 판단
summary(lm_out_count)
# 단계적 선택법
step(lm(Y~, data = data_count), direction = 'both')
# AIC가 가장 작은 회귀모형
lm_out_count_1<-lm(Y ~ X1 + X2 + X5 + X9 + X11 + X12 + X14 + X17 +
X19 + X21 + X22 + X23 + X27 + X28, data = data_count)
lm_out_count_1
# 회귀모형 적합도, 회귀계수 유의성 판단
summary(lm_out_count_1)
```

#### 7.3.5. 비 계층적 군집분석(K-means)(4.1.2)

```
library(readxl) #엑셀파일 로딩을 위한 패키지
library(factoextra) #군집분석 시각화를 위한 패키지

data <- read_excel('결합파일_1095.xlsx') #데이터 로드
data_norm <- sapply(data[,-c(1,2)], scale) #불필요 컬럼 제거 및 데이터 스케일링

set.seed(7777) #난수 고정
km_res_2 <- kmeans(data_norm[,-3],2) #임의의 군집 2개 생성
km_res_3 <- kmeans(data_norm[,-3],3) #임의의 군집 3개 생성
km_res_4 <- kmeans(data_norm[,-3],4) #임의의 군집 4개 생성
km_res_5 <- kmeans(data_norm[,-3],5) #임의의 군집 5개 생성
```

```
fviz_cluster(km_res_2, data_norm[,-3], ellipse.type = 'norm') + theme_minimal() #2군집
시각화
fviz_cluster(km_res_3, data_norm[,-3], ellipse.type = 'norm') + theme_minimal() #3군집
시각화
fviz_cluster(km_res_4, data_norm[,-3], ellipse.type = 'norm') + theme_minimal() #4군집
시각화
fviz_cluster(km_res_5, data_norm[,-3], ellipse.type = 'norm') + theme_minimal() #5군집
시각화

sil2 <- silhouette(km_res_2$cluster, dist(data_norm[,-3])) #2군집의 실루엣계수
sil3 <- silhouette(km_res_3$cluster, dist(data_norm[,-3])) #3군집의 실루엣계수

fviz_silhouette(sil2) #2군집 실루엣계수 시각화
fviz_silhouette(sil3) #3군집 실루엣계수 시각화

##Elbow method로 최적의 군집 수 찾기
tot_withinss <- c()

for(i in 1:20){
  set.seed(1234)
  kmeans_cluster <- kmeans(data_norm[,-3], centers = i, iter.max = 1000)
  tot_withinss[i] <- kmeans_cluster$tot.withinss
}

plot(c(1:20), tot_withinss, type = 'b')
```

#### 7.3.6. 계층적 군집(Hierarchical Clustering)(4.1.3)

```
# 파일 불러오기
df = pd.read_csv('/content/결합파일_1095.csv',encoding='cp949')

# 필요없는 칼럼 삭제 및 범주형 변수 인코딩
df.drop(columns = ['구분'], inplace = True)
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(df['행정동명'])
le_encoded = le.transform(df['행정동명'])
new_cat = pd.DataFrame(le_encoded, columns = ['행정동명_인코딩'])
le.classes_
```

```
# 기존 데이터셋과 병합하고 기존 변수 칼럼 삭제
df = pd.concat([df, new_cat], axis = 1)
df = df.drop(['행정동명'], axis=1)

# 균집화 전 정규화 진행 필수
# 표준 정규화(RobustScaler) 사용: 평균과 분산대신 중간값과 사분위값을 사용하여 이상치 영향 최소화하는 스케일링 기법
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler() # 할당
new_df = scaler.fit_transform(df) # 피팅
df = pd.DataFrame(new_df) # df 변환

# 병합군집(AgglomerativeClustering): 두 인접 클러스터를 반복적으로 병합하는 군집법
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage='ward')
cluster.fit_predict(df)

from scipy.spatial.distance import pdist, squareform
Y = pdist(df)
row_dist = pd.DataFrame(squareform(pdist(df, metric = 'mahalanobis')))
row_dist

from scipy.cluster.hierarchy import linkage
row_clusters = linkage(pdist(df, metric = 'euclidean'),
                       method = 'complete')

# linkage함수에서 초기 데이터 배열을 전달하고, euclidean 지표를 매개변수로 사용
row_clusters = linkage(df.values, metric = 'euclidean', method = 'complete')
# 군집 결과를 데이터프레임 형태로 반환
cl = pd.DataFrame(row_clusters,
                  columns = ['row label 1', 'row label 2', 'distance', 'no. of items in clust.'],
                  index = ['cluster %d' %(i+1) for i in range(row_clusters.shape[0])])
```

### 7.3.7. PyCaret(4.1.4 ~ 4.2)

```
# 데이터 불러오기
df = pd.read_csv('/content/drive/MyDrive/newdata_binom_smote.csv', encoding = 'cp949')
# 필요없는 칼럼 삭제
df.drop(columns = ['탑재형cctv', '고정형cctv','Unnamed: 0'], inplace = True)

# setup으로 설정된 모든 세팅값 확인 및 파라미터 조정
from pycaret.classification import *
```

```
# 파라미터 설명(순서대로): 사용할 데이터, 타겟(종속변수, 구해야하는 값), 불균형 조정(smote)실행, 다중공선성 제거, 수치형 변수 지정(오토모델이라 모델 판단 하에 범주형이면 인코딩 하니까 방지용), fold n번, 훈련/테스트 셋 나눌 때 종속변수의 비율 유지
setup_clf = setup(data= df, target='1095단속유무', fix_imbalance= False,
fix_imbalance_method= None, remove_multicollinearity=True, numeric_features=['주민등록인구','어린이보호구역', '주차구획', '버스정류장', '지하철주요시설', '소화전',
'횡단보도', '주택', '건물_3000', '건물_4000', '건물_5000', '건물_6000', '건물_8000',
'건물_9000', '건물_11000', '건물_14000', '건물_15000', '건물_17000', '건물_18000',
'교차로_교통량_결측치채운버전', '상권_관광여가오락', '상권_부동산', '상권_생활서비스',
', '상권_소매', '상권_스포츠',
'상권_음식', '상권_학문교육'], fold= 5, data_split_stratify= True)

# f1스코어 기준으로 좋은 모델부터 정렬
best = compare_models(sort='F1')

# 위에 돌아간 모델 중에서 특정 모델만 따로 clf에 할당
clf_rf = create_model('rf')

# 시각화
evaluate_model(clf_rf) #plot
interpret_model(clf_rf) # SHAP values
```

### 7.3.8. 총점 계산(4.3.1)

```
# 데이터 불러오기
df = pd.read_csv('전처리데이터.csv', encoding = 'cp949')
점수칼럼 = ['건물_9000', '건물_11000', '건물_14000', '건물_15000', '건물_17000', '건물_18000', '건물_3000',
            '건물_4000', '건물_5000', '건물_6000', '건물_8000', '건물_주택', '관광여가오', '버스정류장',
            '부동산',
            '생활서비스', '소매', '소화전', '스포츠', '어린이보호', '음식', '지하철주요', '학문교육',
            '횡단보도',
            '교차로교통량', '주차구획수','인구수격자']

가중치 = [1] * len(점수칼럼)

# Null 이 있을 수 있는 칼럼은 Null을 0로 변경
pre_process_set.fillna(0, inplace=True)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

```

# 점수칼럼 정규화
norm = pre_process_set.copy()
norm.fillna(0, inplace=True)
norm.set_index('gid', inplace=True)
norm[점수칼럼] = scaler.fit_transform(norm.loc[:,점수칼럼])
norm.reset_index(inplace=True)

# 총점 계산
df = norm.copy()
for i in range(0, len(점수칼럼)):
    name = 점수칼럼[i]
    df[name] = df[name] * 가중치[i]

df.set_index(['gid','lon','lat'], inplace=True)
df['sum'] = df.sum(1)
df.reset_index(inplace=True)

# 등수 계산
df['순위'] = df['sum'].rank(method='dense', ascending=False).astype(int)
final = df.sort_values(by='순위')
final.reset_index(drop=True, inplace=True)

# 상위 20개만 확인
display("결과", final.head(20))

# 최종 결과 CSV 파일쓰기
final.to_csv('결과_총점.csv', index=False, sep=',', encoding='cp949')

# 상위 20개 구역 지도 시각화
import folium

for i in range(20):
    지도보기 = folium.Map(location=[final.loc[i,'lat'],final.loc[i,'lon']], zoom_start = 15)
    folium.Marker([final.loc[i,'lat'],final.loc[i,'lon']]).add_to(지도보기)
    print (final.loc[i,'lat'],final.loc[i,'lon'])
    j = i+1
    display(f'상위 {j}번째 격자의 센터 위치 보기', 지도보기)

```

## 7.3.9. 데이터 변환 및 오버샘플링(4.1.4)

```

library(readxl) #엑셀파일 로딩을 위한 패키지
library(DMwR) #SMOTE 오버샘플링을 위한 패키지
library(smotefamily) #ADASYN 오버샘플링을 위한 패키지
library(dplyr) #R 데이터프레임 편집 패키지

data <- read_excel('결합파일_1095.xlsx') #데이터 로드
data_binom <- data #데이터 복사
data_binom$단속지점개수 <- ifelse(data$단속지점개수 == 0, 0, 1) #단속지점개수의 컬럼의
값 변경
data_binom$단속지점개수 <- as.factor(data_binom$단속지점개수) #범주형 종속변수 변환(0
과 1)

newdata_binom <- data_binom[-c(1,2,5)] #불필요한 컬럼 제거
newdata_binom <- newdata_binom %>% rename('1095단속유무' = '단속지점개수') #변
수명 변환
newdata_binom <- as.data.frame(newdata_binom) #데이터프레임으로 구조 변환

newdata_binom_smote <- SMOTE('1095단속유무' ~ ., newdata_binom, perc.over = 300,
perc.under = 135, k = 5) #smote 오버샘플링

rownames(newdata_binom_smote) = NULL #행 이름 제거

newdata_binom_adas <- ADAS(newdata_binom[,-28], newdata_binom[,28], K = 5)
#ADASYN 오버샘플링

newdata_binom_adas <- newdata_binom_adas$data #ADASYN 샘플링 결과 추출
newdata_binom_adas$class <- as.factor(newdata_binom_adas$class)
#범주형 종속변수 추가

write.csv(newdata_binom_adas,'newdata_binom_adas.csv', fileEncoding = 'cp949')
#CSV로 저장
write.csv(newdata_binom_smote, 'newdata_binom_smote.csv', fileEncoding = 'cp949')
#CSV로 저장

```