

데이터기반 행정으로 국민의 삶의질을 개선하라!  
데이턴십 해커톤 제 4회

# 화재 위험 예측 모델을 이용한 소방 활동 지원대책 마련

분석 결과보고서

참여조: 17조

참여자: 구희연(조장)

김강현

김예린

임선아

정수빈

조범진

씨에스리 컨소시엄

CSLEE kpc 한국생산성본부

## 목 차

1. 분석 개요 .....	3p
1.1. 분석 배경 및 개요 .....	3
1.2. 분석 목적 및 방향 .....	6
1.3. 분석 결과 활용 방안 .....	6
2. 분석 데이터 .....	8p
2.1. 분석 데이터 목록 .....	8
2.2. 데이터 상세 설명 .....	9
2.3. 데이터 정제 방안 .....	10
3. 분석 프로세스 .....	15p
3.1. 분석 프로세스 .....	15
3.2. 분석 내용 및 방법 .....	16
4. 분석결과 .....	20p
4.1. 분석 프로세스 .....	20
4.2. 등급 산출을 위한 모델링 .....	21
4.3. CCTV 설치 지역 선정 및 소방용수시설 점검 지역 선정 시각화 .....	23
5. 활용 방안 .....	30P
5.1. 문제점 개선 방안 .....	30
5.2. 업무 활용 방안 .....	30
6. 참고자료(Reference) .....	31P
7. 부록 .....	32P
7.1. 주제설계를 위한 마인드맵 .....	32
7.2. 분석 상세코드 .....	33

## 1. 분석 개요

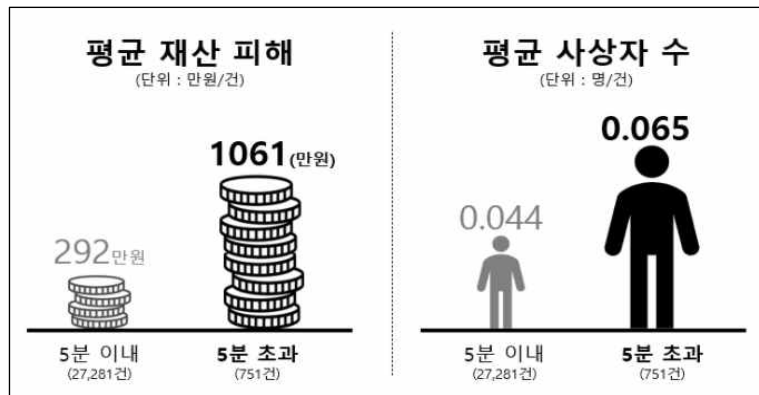
### 1.1 분석 배경 및 개요

#### 1.1.1. 분석 배경

##### 1) 화재진압 골든타임

:인명 구조나 화재 진압의 사고 초기 대응에 있어서 가장 중요한 시간

- 소방청에 의하면 화재 진압의 골든타임은 5분이며, 플래시 오버(Flash over, 폭발적으로 불이 번지는 것)까지 약 8분 소요
- 국가화재정보시스템에 따르면, 2016년 서울에서 발생한 총 6443건의 화재 사건 중 소방차가 5분 이후에 도착한 경우가 총 928건으로 골든타임을 지키지 못한 비율이 14.4%로 집계
- 골든타임 초과 시 평균 재산 피해액은 1,061만 원으로 준수하였을 때 보다 3.6배 많았고, 사상자는 49명으로 1.5배 많음



[그림 1-1] 소방출동대의 현장 도착 시간 대비 피해 현황 (서울연구원)

- 불법 주정차 및 예측 불가능한 도로 상황 등의 제약요인으로 인해 매년 소방차가 골든타임 이내 도착 비율은 2013년부터 2017년까지 5년 동안 60% 전후에 그침

Year	Total	≤3mins	≤5mins	≤10mins	≥20mins	Rate of arrival within golden time
2017	44,178	7,601	17,854	12,399	6,324	58%
2016	43,413	7,715	18,069	11,798	5,831	59%
2015	44,435	7,923	19,576	10,985	5,951	62%
2014	42,135	8,088	17,604	10,291	6,152	61%
2013	44,178	8,157	15,429	11,609	5,737	58%

[그림 1-2] 현장 도착시간대별 화재건수 (National Fire Agency, 2017)

#### 2) 상태 불량 소방시설로 인한 초기 골든타임 초과

- '소방시설이란 소화 설비, 경보 설비, 피난구조 설비, 소화용수설비, 소화 활동설비를 말함 ('화재예방, 소방시설 설치·유지 및 안전 관리에 관한 법률' 제 2 조 제 1 항 제 1 호)

[표 1-1] 소화 설비의 종류

종류	세부종류
소화기구	소화기, 간이소화용구, 자동확산소화기
자동소화장치	주거용·상업용·캐비닛형·가스·분말·고체애어로졸 자동소화장치
옥내소화전설비	(호스릴옥내소화전설비 포함)
스프링클러설비등	스프링클러설비, 간이스프링클러설비(캐비닛형 간이 스프링클러설비 포함), 화재 조기진압용 스프링클러 설비
물분무등소화설비	물 분무 소화설비(이하 "소화설비" 생략), 미분무, 포, 이산화탄소, 할론, 할로겐 화합물 및 불활성 기체(다른 원소와 화학 반응을 일으키기 어려운 기체를 말함), 분말, 강화액, 고체애어로졸 소화설비
옥외소화전설비	-

(「화재예방, 소방시설 설치·유지 및 안전 관리에 관한 법률」 제 2 조 제 1 호, 「화재예방, 소방시설 설치·유지 및 안전 관리에 관한 법률 시행령」 별표 1 제 1 호)

- 소방시설 관리가 미흡하여 신속한 소방 활동 및 소방용수 확보가 지연될 경우, 지체된 시간만큼 피해 규모가 증가함
- 실제 화재 현장에 설치돼 있는 소방용수시설에서 물이 나오지 않아 초기 화재진압의 골든타임을 놓치면 대형 화재로 이어질 수 있음

#### [관련 자료]

- '소방용수시설 고장 현황' 자료에 따르면 2016년 기준 전국 소방용수시설 고장건수는 4338건으로 2013년 대비 고장률이 94% 증가함
- 지역별 고장 증가율은 세종(733%), 경기(235%), 전남(233%), 서울(213%), 충남(145%), 부산(103%), 경북(76%), 울산(41%), 경남(32%) 순으로 나타남
- 2020년 상반기 소방용수시설 고장 건수가 4961건을 기록하며 매년 고장 증가율이 상승하고 있는 걸 알 수 있음

## 3) 불법 주정차로 인한 소방차 진입로 확보 문제

- ☐ 초기 화재 대응을 위해서는 소방로 확보가 중요함
- ☐ 실제로 소방공무원 102명 중 41%는 신속한 화재 진압에 가장 큰 장애요인으로 '소방차 진입이나 배치 등 소방 활동 공간 협소 문제'를 지적함 (주) '화재취약지구의 화재안전성 개선방안 연구', '한국방재학회 논문집' (제15권 5호))
- ☐ 소방청에 문의한 결과, 소형 소방차의 폭은 약 2.5m로 통행을 위해 최소 폭 3m 도로 확보가 필요하나, 도로 좌우에 불법주차 차량이 한 대라도 있으면 진입 불가능하고 대형 소방차의 폭은 약 6m로 통행을 위해 최소 폭 10m 도로 확보가 필요
- ☐ 서울시에서 실시한 2015년 조사 결과, 소방차가 진입하기 힘든 곳이 총 1183곳
- ☐ 소방차 진입이 어려운 지역의 60%는 주거지역으로 화재 발생 시 인명피해 가능성이 높음

## [관련 사례]

## 1) 2015년 의정부 화재

- 2015년 1월 10일, 경기도 의정부시 시내에 위치한 주거용 오피스텔 건물에서 화재가 일어나 약 130여 명의 사상자가 발생한 사건
- 해당 사건의 초기 대응 실패 원인으로 부실한 소방시설, 화재에 취약한 건물 구조 등의 문제점이 있었으나 가장 큰 원인은 불법 주정차 차량으로 인한 소방차 진입로 확보 실패

## 2) 2016년 쌍문동 화재

- 2016년 9월 24일, 서울 도봉구 쌍문동의 고층 아파트에서 화재가 일어나 일가족 3명이 숨지고 17명의 부상자 발생
- 해당 사건은 '소방차 불통' 지역의 문제를 그대로 보여준 사건
- 당시 소방차가 불법 주정차 차량으로 인하여 현장 도착 지연

## 3) 2017년 제천 스포츠센터 화재

- 2017년 12월 21일, 충청북도 제천시 하소동에 있는 스포츠센터에서 화재가 일어나 65명의 사상자 발생
- 가연성 외부 마감재로 순식간에 불길이 번졌고 소방당국은 인명 피해가 컸던 이유로 도로 양쪽에 있던 불법주차 차량으로 인해 먼 거리를 우회하거나 견인차로 차량을 치우느라 현장 진입에 30분 이상 지연됨을 지적함

## 1.1.2. 필요성

## 1) 연구 객관성 확보

- ☐ 소방청 '화재 위험 평가 지표 연구 보고서'에 따르면, 화재 위험도 연구는 사람이 직접 조사하는 방식으로 진행된다. 이는 당시 현장에 파견된 조사관의 주관성 개입의 우려가 있으므로, 보다 객관적이고 정확한 조사를 위해 데이터 기반의 화재 위험 예측 모형 개발이 필요
- 2) 위험 예측을 통한 소방 활동의 선제적 지원
  - ☐ 지역별 화재 위험도에 따라 화재진압 골든타임 확보를 위한 차등적 소방 활동 지원책을 수립할 필요가 있음
  - ☐ 고위험 지역에 선제적 지원 대책으로 소방시설의 우선 점검 지점 및 불법 주정차 단속 CCTV 설치 지점 선정이 있음

## 1.2. 분석 목적 및 방향

## 1.2.1. 분석 목적

- ☐ 기계학습을 통한 데이터 기반의 객관적이고 정확한 화재 취약지역 분석
- ☐ 화재 진압 골든타임을 확보하기 위하여 화재취약지역의 소방용수시설 및 불법 주정차 단속 CCTV의 현황 분석을 통한 선제적 안전대책 마련

## 1.2.2. 분석 방향

- ☐ 차세대 화재 위험 예측 모델은 서울특별시를 대상으로 지역 데이터를 활용하여 지역별 화재 위험 예측 및 예방을 위함
- ☐ 화재 예방, 소방시설, 피난 능력, 건축 방재의 범주로 나누어 화재 위험 주요인을 선정하고 변수의 중요도를 설정하여 새로운 화재 위험 평가 지표를 제시
- ☐ 화재 위험 평가 지표를 활용하여 위험 등급 산출을 위한 기준 수립 및 위험 지수를 산출
- ☐ 화재 위험 지수를 바탕으로 지역별 화재 위험 지역을 10등급으로 분류
- ☐ 화재 위험 및 취약 지역 선정 및 기존 안전대책의 한계를 보완한 맞춤형 소방 활동 지원 대책 수립(점검 우선 지역 선정, CCTV 설치 지점 선정)

## 1.3 분석 결과 활용 방안

## 1.3.1. 화재 위험 등급 모델 활용

- ☐ 빅데이터에 근거한 정량적 화재 위험성 평가
- ☐ 시구 및 행정동 별, 화재 고위험 등급 지도 시각화
- ☐ 화재 고위험 등급 지도를 기반으로 한 화재 취약 지역 선정 및 화재 진압 골든타임 확보를 위한 맞춤형 화재 대응 전략 수립

## 1.3.2. 소방 활동 지원 대책 제안

- ☐ 화재 고위험 등급에 속하는 지역 내에 소화용수시설 및 불법 주정차 단속 CCTV 위치 시각화를 통한 정밀 점검 지점 선정
- ☐ 소방차 진입 불가한 구역에 위치한 소화용수시설 안전 점검 우선 실시 대책 수립
- ☐ 불법 주정차가 있을 시, 소방차가 진입이 어려운 도로가 있는 지역 중에 소화용수시설 밀집된 곳을 중점으로 불법 주정차 단속 CCTV 우선 설치 지점 선정

## 2. 분석 데이터

## 2.1. 분석 데이터 목록

[표 2-1] 분석 데이터 목록

활용 데이터	구분	필수(Y/N)	생성 주기	수집 유형	활용 방법
소방청 화재발생 주조정보	정형/ 내부	Y	실시간	Excel	- 국가 화재 정보 시스템 화재 현황 ('2017. 4만4천건) - 일 단위 집계 - 실시간 수집
서울시 소화용수 위치정보	정형/ 외부	N	매년	Excel	- 서울시 소화용수시설 위치 좌표 (x_layer, y_layer)
신종업종 화재위험 평가지표 및 위험지수 개발연구 보고서	비정형/ 내부	Y	일회성	Pdf	- 위험지수 개발 및 평가등급 산정 (보고서-텍스트)
서울 생활인구(내국인)	정형/ 외부	N	실시간	Excel	- 성별/연령대별 생활인구수 (총생활인구수, 남/여 생활인구수)
서울특별시 불법주정차/ 전용차로 위반 단속CCTV 위치정보	정형/ 외부	Y	분기	Excel	- 서울시 불법주정차 단속 CCTV 위치 좌표 (위도, 경도)
서울 행정구역 데이터	비정형/ 내부	Y	매년	Shp	- 서울시 행정구역 좌표
건축물 맞춤형 화재위험 평가기술 개발	비정형/ 외부	N	일회성	Pdf	- 위험지수 개발 및 평가등급 산정 (보고서-텍스트)
서울시 도로 폭 데이터	비정형/ 내부	Y	수시	Shp	- 도로 폭을 기반으로 주변 용수시설 및 불법주정차 단속 CCTV 현황 분석

## 2.2 데이터 상세 설명

### 1) 수급 데이터

[표 2-2] 수급 데이터 목록

분석 데이터명		컬럼명	데이터타입	필수(Y/N)	제공기관
소방청 화재발생 주소정보	주소정보	시도	Char	Y	소방청
		시군구			
		지번동			
	화재예방	소방서거리	float	Y	
		안전센터거리			
		다중이용업여부	Bool		
		방화관리대상여부			
	소방시설	자동화재감지기	Char	Y	
		자동소화설비 (작동및효과성)			
		소화기구(사용여부)	Bool		
		옥내소화전(사용여부)			
	피난시설	건물층수(지상)	float	Y	
		건물층수(지하)			
		바닥면적	Char		
유도등					
비상경보설비					
비상방송설비					
건축방재	건물구조조	Char	Y		
	창소충분류				
서울 생활인구(내국인)		기준일	Date	N	서울특별시
		행정동코드			
		총생활인구수			
신종업종 화재위험 평가지표 및 위험지수 개발연구 보고서		화재위험지수 안전등급표	Txt	Y	소방청
건축물 맞춤형 화재위험 평가기술 개발		화재안전성능 평가표	txt	Y	한국건설기술연구원

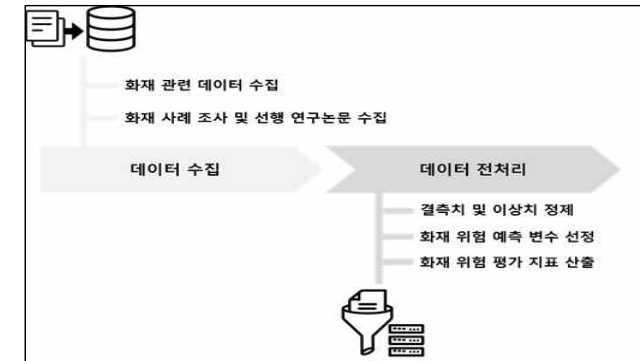
### 2) GIS 데이터

[표 2-3] GIS 데이터 목록

분석 GIS	데이터명	컬럼명	데이터타입	필수(Y/N)	제공기관
소방용수시설	서울시 소화용수 위치정보	x_layer	Float	Y	서울종합방재센터
		y_layer	Float	Y	
GIS	서울 행정구역 데이터	읍/면/동 경계에 대한 공간정보 및 속성정보	Shp	Y	통계청
불법주정차 단속 GIS	서울특별시 불법 주정차/ 전용차로 위반 단속 CCTV 위치정보	고정형 CCTV 지번주소	Float	Y	서울특별시
		위도	Float	Y	
		경도	Float	Y	
		자치구	Char	Y	
		단속지점명	Char	N	
서울시 도로 폭 현황 데이터	서울시 도로 폭 현황 데이터	현장구분	Char	N	한국지역정보개발원
		ROAD_BT	Shp	Y	

## 2.3 데이터 정제 방안

### 2.3.1. 데이터 전처리 프로세스



[그림 2-1] 데이터 전처리 프로세스

### 2.3.2. 변수 선정

- 화재 발생 위험도에 대한 정량적 평가 모델을 개발하기 위해 소방청에서 제공하는 '신종업종 화재 위험 평가 지표 및 위험지수 개발 연구 결과 보고서'를 참고하여 예측 변수를 선정함
- 화재 발생 이후 데이터를 사용할 경우, 일반 건물의 정보로만 화재 취약성을 판단할 수 없는 논리적 오류가 발생하기 때문에 위험 요소를 줄이고자 기존 측정지표를 다음과 같이 수정함

[표 2-4] 변수 선정 결과

측정영역	측정치표	선택 변수
화재예방	피해확대위험	소방서거리
		안전센터거리
		다중이용업소여부
		방화관리대상여부
소방시설	화재감지시스템	자동화재감지기
	소화설비	자동소화설비
		소화기구
		옥내소화전
피난시설	영업장 내부통로와 출구 피난 능력	층수(지상)
		층수(지하)
		면적
	피난안내설비	유도등
		비상경보설비
		비상방송설비
건축방재	내화구조 및 마감재료	건물구조조
	구획 및 비상구	창소충분류

## 2.3.2. 데이터 정제

[표 2-5] 데이터 정제(선택변수 및 세부 내용)

사용유형	측정영역	측정지표	선택변수	단위 및 세부 내용
분류	시도			서울특별시, 경상북도, 경기도 ...
	시군구			강남구, 문경시, 거제시 ...
	읍면동			논현동, 홍덕동, 일운면 ...
	화재유형			건축, 임야, 자동차 ...
등급산출	화재예방	피해확대위험	소방서거리	수치형 - km
			안전센터거리	수치형 - km
			다중이용업소여부	Y, N, (null)
			방화관리대상여부	Y, N, (null)
	소방시설	화재감지시스템	자동화재감지기	미상, 열감지기, 기타 감지기 ...
			자동소화설비	효과적작동, 미상, 미작동 ...
			소화기구	사용, 기타 소화기구, 노후화 ...
			옥내소화전	사용, 설치 불량, 미상 ...
	피난시설	영업장 내부통로와 출구 피난 능력	층수(지상)	수치형
			층수(지하)	수치형
			면적	수치형 - m <sup>2</sup>
		피난안내설비	유도등	작동, 미상, 전구불량 ...
			비상경보설비	경보, 미상, 고장, 전원차단 ...
			비상방송설비	경보, 미상, 고장, 전원차단 ...
	건축방재	내화구조 및 마감재	건물구조조	목조, 벽돌조, 철근조 ...
		구획 및 비상구	장소중분류	공연장, 음식점, 카페 ...

## 1) 화재예방

- 피해 확대 위험: 소방서 거리, 안전 센터 거리의 km 단위를 0~1로 스케일링하고, 다중이용업소 여부, 방화관리 대상 여부 칼럼의 Yes / No를 0, 1로 정제하여 사용함

[표 2-6] 화재예방 데이터 정제

변수	원본 값	수치로 변환된 값
다중이용업여부	N, (null)	0
	Y	1
방화관리대상여부	N	0
	Y	1

## 2) 소방시설

- 화재 감지 시스템의 적절성 및 적응성 평가: 자동 화재감지기의 작동 유무에 따라 0과 1로 정제함
- 화재 감지 시스템의 적절성 및 적응성 평가: 자동 화재감지기의 작동 유무에 따라 0과 1로 정제함

[표 2-7] 소방시설 데이터 정제

변수	원본 값	수치로 변환된 값
자동화재감지기	불꽃감지기, 연기감지기, 열감지기, 열연복합식 감지기	0
	미상, (null)	1
자동소화설비	효과적 작동	0
	미상, 효과없음(관리소홀, 기타, 미상), 미작동, (null)	1
소화기구	사용	0
	고장, 기타 소화기구활용, 노후, 미상, 사용법 미숙지, 소화약제 부족, (null)	1
옥내소화전	사용, 기타 옥내소화전	0
	미상, 방수압력 미달, 설비불량, 사용법 미숙지, (null)	1

## 3) 피난능력

- 영업장 내부 통로의 형태와 출구의 피난능력 평가: 지상층(층수)은 바로 스케일링하였고, 지하층(층수)은 한국 화재소방학회 '지하공간 방재대책에 대한 연구'에 따라 2배의 중요도를 부여하여 면적(m<sup>2</sup>) 단위를 0~1로 스케일링함
- 피난안내설비 측정: 유도등, 비상경보 설비, 비상방송설비의 작동 유무를 0과 1로 정제함

[표 2-8] 피난능력 데이터 정제

변수	원본 값	수치로 변환된 값
유도등	작동, 기타	0
	미상, (null)	1
비상경보설비	경보	0
	기타, 미상, 사용법 미숙지, 수신기전원차단, 음향장치고장, (null)	1
비상방송설비	방송	0
	기타, 미상, 전원차단, (null)	1

## 4) 건축 방재

- 내화구조 및 마감재로 평가: 건물 구조 조(마감재) 중 화재 발생 위험이 높은 목조, 비닐하우스 파이프 조, 샌드위치 패널 조를 1로 설정하고, 화재 위험이 낮은 구조 조를 0으로 설정함
- 구획 및 비상구 평가: 장 소중 분류의 주거시설, 업무시설 등 다중이용업, 방화관리 대상 여부에 따라 항목을 0과 1로 정제함

[표 2-9] 건축방재 데이터 정제

변수	원본 값	수치로 변환된 값
건물구조조	기타 조, 벽돌조, 블록조, 철골조, 철골철근콘크리트조, 철근콘크리트조, 철조, 지장벽돌조, 컨테이너조	0
	목조, 비닐하우스 파이프조, 샌드위치패널조	1
장소중분류	건강시설, 공공기관, 공공주택, 공연장, 공장시설, 관람장, 기타건축물, 기타주택, 노유자시설, 단독주택, 동식물시설, 숙박시설, 역사터미널, 연구학원, 운동시설, 음식점, 의료시설, 일반업무, 자동차시설, 작업장, 전시장, 종교시설, 지중시설, 창고시설, 청소년시설, 판매시설, 학교, 항공시설	0
	오락시설, 위락시설, 일상서비스	1

## 5) 전처리 전/후 비교

시도	시군구	읍면동	소방서 거리 (km)	안전센터 타거리	다중 이용 여부	방화관 리대상 여부	자동차 화재 감지 기	자동차 소화 설비(작동 및과장)	소화기구 (사용/미 사용여부)	옥내소화 전(사용/미 사용여부)	건물 층수 (지상)	건물 층수 (지하)	바닥면 적	유 도 경로 설비	비상 방송 설비	건물 구조 조	장소 중분 류
0	서울 특별시	강남구 논현동	4.1	3.0	N	Y	열감 지기	미설(미설)	미설	미설	6.0	1.0	168.00	미설	미설	철골 철근콘 크리트 조	건강 시설
1	강남 북도	동작 동	2.5	2.5	N	N	미설	미설(미설)	미설	미설	1.0	0.0	80.40	미 설	미설	벽돌 조	일반 업무
2	경기 도	안산 시정 북부	5.9	2.9	N	N	미설	미설(미설)	미설	미설	3.0	1.0	139.24	미 설	미설	철골 철근콘 크리트 조	운동 시설
3	전라 북도	익산 시 인화 동1가	7.5	1.0	N	Y	열감 지기	미설(미설)	사용	기타 옥내 소화전 설비	8.0	0.0	268.11	작 동	미설	철골 철근콘 크리트 조	숙박 시설
4	경기 도	하남 시 합동 동	5.5	3.3	N	N	미설	미설(미설)	미설	미설	4.0	0.0	130.00	미 설	미설	철골 철근콘 크리트 조	건강 시설

[그림 2-2] 전처리 전 데이터 Set

index	시도	시군구	읍면동	소방서 거리 (km)	안전센터 타거리	다중 이용 여부	방화 관리 대상 여부	자동차 화재 감지 기	자동차 소화 설비(작동 및과장)	소화기구 (사용/미 사용여부)	옥내소화 전(사용/미 사용여부)	건물 층수 (지상)	건물 층수 (지하)	바닥면 적	유 도 경로 설비	비상 방송 설비	건물 구조 조	장소 중분 류	
0	서울 특별시	강남구	논현동	4.1	3.0	0	1	0	1	1	1	6.0	1.5	168.00	1	1	1	0	0
1	강남 북도	동작시	동작동	2.5	2.5	0	0	1	1	1	1	1.0	0.0	80.40	1	1	1	0	0
2	3	경기도	안산시 시정 북부	5.9	2.9	0	0	1	1	1	1	3.0	1.5	139.24	1	1	1	0	0
3	4	전라 북도	익산시 인화 동1가	7.5	1.0	0	1	0	1	0	0	8.0	0.0	268.11	0	1	1	0	0
4	5	경기도	하남시 합동동	5.5	3.3	0	0	1	1	1	1	4.0	0.0	130.00	1	1	1	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

[그림 2-3] 전처리 후 데이터 Set

## 2.3.3 변수별 중요도 부여 방법

- '[참고자료 1] 신종업종 화재 위험 평가 지표 위험지수'와 '[참고자료 2] 화재안전성능지표'의 평가 배점을 참고하여 건물 및 지역에 대한 종합적인 화재안전 성능 평가 기준을 다음과 같이 마련함

[표 2-10] 화재안전 성능 평가 기준

평가요소	중요도	평가요소	해당컬럼	중요도	평가지표 위험지수
화재예방	0.242	피해확대위험	소방서거리 안전센터거리 다중이용업소여부 방화관리대상여부	0.242	24.2%
		소계(합계)		0.242	
소방시설	0.208	화재감지시스템	자동화재감지기	0.137	13.7%
		소화설비	자동소화설비 소화기구 옥내소화전	0.071	7.1%
		소계(합계)		0.208	
피난능력	0.371	영업장 내부통로와 출구 피난능력	층수(지상) 층수(지하) 면적	0.206	20.6%
		피난안내설비	유도등 비상경보설비 비상방송설비	0.165	16.5%
		소계(합계)		0.371	
건축방재	0.179	내화구조 및 마감재료	건물구조조	0.08	8%
		구획 및 비상구	장소중분류	0.099	9.9%
		소계(합계)		0.179	
합계	1.000			1.000	100%

## 2.3.4 위험등급 산출

- 위험지수 산출하고 이를 기반으로 목표 변수인 위험 등급 산출
- 전처리를 통해 수치로 변환된 데이터를 표준화하는데 sklearn 패키지 중에 MinMaxScaler를 사용하여 0~1 사이의 값으로 변환

```

In [9]: #표준화를 사용하기 위한 sklearn 사용
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()

feature3 = ["소방서거리(km)", "안전센터거리", "다중이용업소여부", "방화관리대상여부", "자동차화재감지기",
            "자동소화설비(작동및과장)", "소화기구(사용/미사용여부)", "옥내소화전(사용/미사용여부)",
            "건물층수(지상)", "건물층수(지하)", "바닥면적", "유도등", "비상경보설비", "비상방송설비", "건물구조조", "장소중분류"]
new_df3 = df_wrac2(feature3)

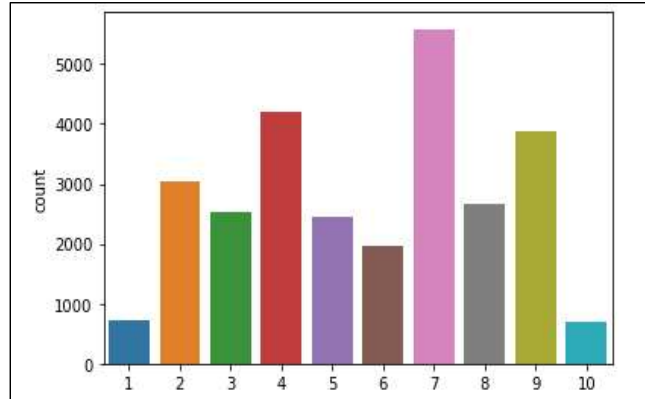
#표준화 객체
new_df3 = sc.fit_transform(new_df3)
new_df3 = pd.DataFrame(new_df3)

#결과 이름 지정
new_df3.columns = ["소방서거리", "안전센터거리", "다중이용업소여부", "방화관리대상여부", "자동차화재감지기",
                  "자동소화설비", "소화기구", "옥내소화전", "건물층수(지상)", "건물층수(지하)", "바닥면적",
                  "유도등", "비상경보설비", "비상방송설비", "건물구조조", "장소중분류"]

```

[그림 2-4] 위험등급 산출 코드

- 산출한 위험지수를 바탕으로 등급을 10개로 분할
- 등급을 분할하는 기준은 '[그림 2-12] 화재안전 성능 평가 기준'으로 함
- 아래의 그래프는 1~10으로 등급을 나눈 것을 카운팅 하여 시각화한 것

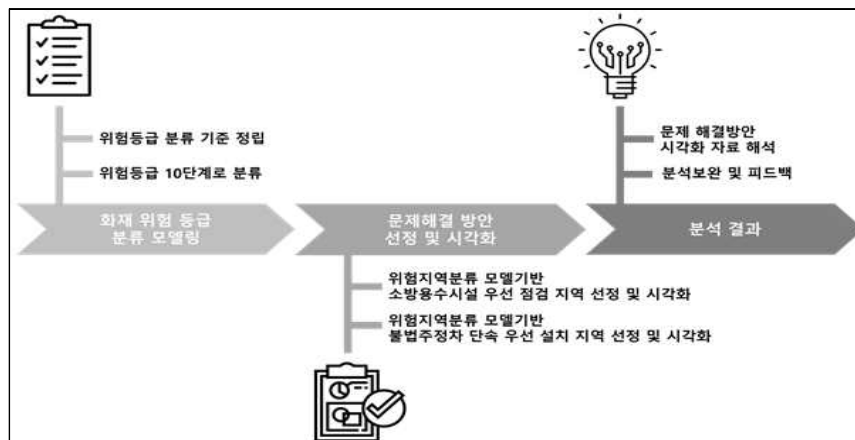


[그림 2-5] 화재안전 등급 그래프

- 목표 변수를 만들고 분석 모델링 진행
- 앞서 산출한 위험 등급을 기반으로 모델의 정확도를 평가함

### 3. 분석 프로세스

#### 3.1. 분석 프로세스



[그림 3-1] 분석 프로세스

### 3.2 분석 내용 및 방법

#### 3.2.1. 학습 / 검증 데이터 나누기

##### □ 학습 데이터

- 서울특별시를 제외한 전국 모든 데이터를 학습 데이터로 사용
- 사용된 데이터는 22951행

##### □ 검증 데이터

- 서울특별시만 검증 데이터로 사용
- 사용된 데이터는 4758행

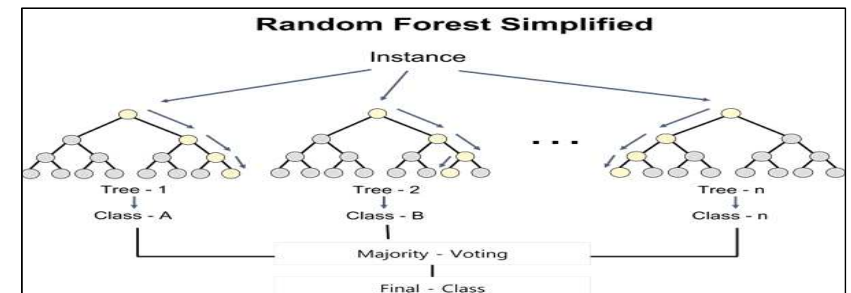
##### □ 목표 변수

- 목표 변수
- 사용된 데이터는 4758행

#### 3.2.2. 분석 알고리즘 소개

##### □ Random Forest Classification

- 기본 결정 트리(Decision Tree)는 해당 데이터에 대한 맞춤 분류를 진행하기 때문에 과적합 현상(Overfitting)이 자주 나타남
- 배기(Bagging) 머신러닝 모델 중 하나이며, 여러 개의 decision tree를 형성하고 새로운 데이터 포인트를 트리에 동시에 통과시키며 투표를 실시하여 가장 많이 득표한 결과를 최종 분류 결과로 선택하는 알고리즘
- 기본 결정 트리에서 문제가 되었던 과적합 현상을 앙상블 기법으로 보완한 모델
- 각각의 트리가 독립적으로 학습하기 때문에 학습과정을 병렬화 가능
- 모델 내부에 트리를 만들기 위해 데이터에 대한 부트스트랩 샘플(중복을 허용한 샘플)을 생성
- 생성 데이터 샘플을 기반으로 트리들을 생성
- 몇 개의 특성(max\_features)을 고를지, 몇 개의 트리(n\_estimators)를 만들지 등을 파라미터로 받아 생성



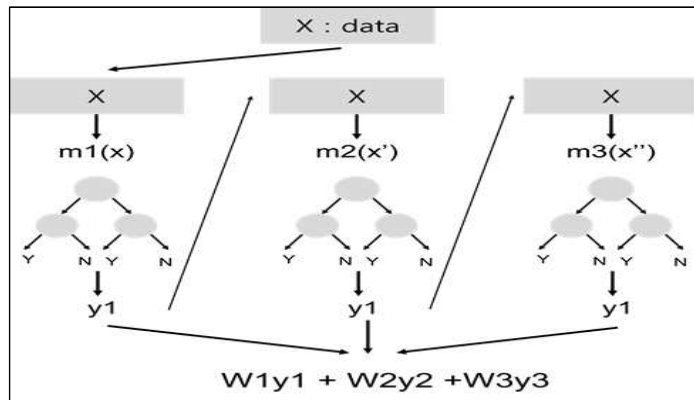
[그림 3-2] 랜덤포레스트 알고리즘



### □ XGBoost Classification

- XGBoost는 Gradient Boosting 방법 중 한 가지
- 설명에서처럼 효율성과 유연성, 휴대성(efficient, flexible, portable)이 뛰어남
- 유연한 Learning 시스템 (여러 파라미터를 조절해가면서 최적의 Model을 만들 수 있음)
- 신경망에 비해 시각화가 쉽고 직관적
- 자원(CPU, 메모리)이 많을수록 빠르게 학습하고 예측 가능
- Cross validation을 지원함
- GBM 알고리즘에 정규화식을 추가한 알고리즘으로 CART 기반으로 분류와 회귀 둘 다 가능하다는 특징이 있음
- 주요 특징

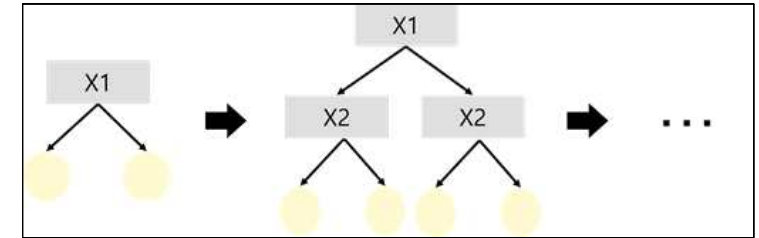
1. GBM 대비 빠른 수행 시간을 가짐
2. 과적합 규제한 기능을 구현함
3. 결측치를 내부적으로 처리함
4. 나무 가지 치기를 통해 분할 수를 더 줄임



[그림 3-3] XGBoost 알고리즘

### □ CatBoost Classification

- CatBoost의 약자는 Categorical Boosting
- Categorical feature를 처리하는 데 중점을 둔 알고리즘
- CatBoost는 GBM을 구현해놓은 패키지 중에 하나
- CatBoost는 기존의 부스팅 과정과 전체적인 양상은 비슷하되, 조금 다름.
- 기존의 부스팅 모델이 일괄적으로 모든 훈련 데이터를 대상으로 잔차 계산을 했다면, CatBoost는 일부만 가지고 잔차 계산을 하여 모델을 만들고 해당 모델로 예측한 값을 데이터의 잔차로 사용



[그림 3-4] CatBoost 알고리즘

### 3.2.3 분석 방법

- 1) 공공기관에서 제공하는 공공데이터 수집 및 정제
  - 화재 발생 정보, 소화용수시설 위치정보, 생활인구, 건물 등 수집한 데이터 셋의 결측치 및 이상치를 정제
- 2) 새로운 화재 위험 평가 지표 제시
  - 화재 예방, 소방시설, 피난 능력, 건축 방재의 범주로 나누어 화재 위험 주요인을 선정하고 변수의 중요도를 설정

### 3) 화재 위험 등급 분류를 위한 화재 위험식 제시

- 새로운 화재 위험 평가 지표 활용

$$Y = 0.242 \times a + 0.137 \times b + 0.071 \times c + 0.206 \times d + 0.165 \times e + 0.08 \times f + 0.099 \times g$$

( Y=화재위험식,

a=피해확대위험,

b=화재감지시스템, c=소화설비,

d=영업장내부통로와 출입구의 피난능력, e= 피난안내설비,

f=내화구조 및 마감재료, g= 구획 및 비상구 )

- 4) 전처리 과정을 마친 데이터에 위험식을 대입하여 지역별 화재 위험지수를 산출

- 5) '[그림 2-4] 화재안전 성능 평가 기준'을 참조하여 화재 안전 성능 평가 결과 분류 기준 수립하여 위험지수를 등급(1~10)으로 분류

- 6) 산출된 등급을 Target으로 Classification 모델을 구동시켜 화재 위험등급의 당위성을 확보

- 7) 지역별 화재 위험 등급 결과를 활용하여 화재 고위험 지역 시각화와

## 화재 예방 모범 지역 시각화

8) 지역별 화재 위험 등급(1~740)을 20%씩 분류하여 A(매우 우수), B(우수), C(보통), D(미흡), E(매우 미흡)로 산정

9) 화재 예방 모범 지역으로 A(1~148)에 해당하는 일부 지역을 추출하여 시각화

10) 화재 위험 등급이 가장 높은 3개의 지역구를 고위험 지역으로 선정하고 해당 구 내 동별 위험지수 시각화

11) 화재 위험 지역의 소방용수시설, 불법 주정차 단속 CCTV 데이터를 분석하여 시각화하고 화재 진압 골든타임을 확보할 수 있도록 지원하는 방안을 모색

12) 고위험 지역 내 소방용수시설 밀집 지역 산출 후 도로 폭에 따라 소방차 진입 불가 지역, 불법 주정차 유무에 따라 소방차 진입 제한지역 산출

- (- 소형 소방차 통행 최소 도로폭 : 3m
- 대형 소방차의 통행 최소 도로폭 : 6m
- 불법 주정차(한 대당 폭 2m 계산, 도로 양쪽 주차 시 4m)
- 불법 주정차가 도로 양쪽에 있을 때, 가장 큰 소방차가 통행하려면 도로 폭이 10m 이상 되어야 함)

13) 불법 주정차로 소방차 이동 제한 도로(도로 폭 3m~10m) 중 소방용수시설 밀집 지역에 불법 주정차 CCTV 우선 설치 지역 선정

14) 소방차 진입 불가 도로(도로 폭 3m 미만) 중 소방용수시설 밀집 지역 우선 점검 지역 선정

15) 결과 비교를 위해 선정한 화재 예방 모범 지역에 소방용수시설, 불법 주정차 단속 CCTV 데이터를 병합하여 시각화하고 일부 부분 추출

## 4. 분석 결과

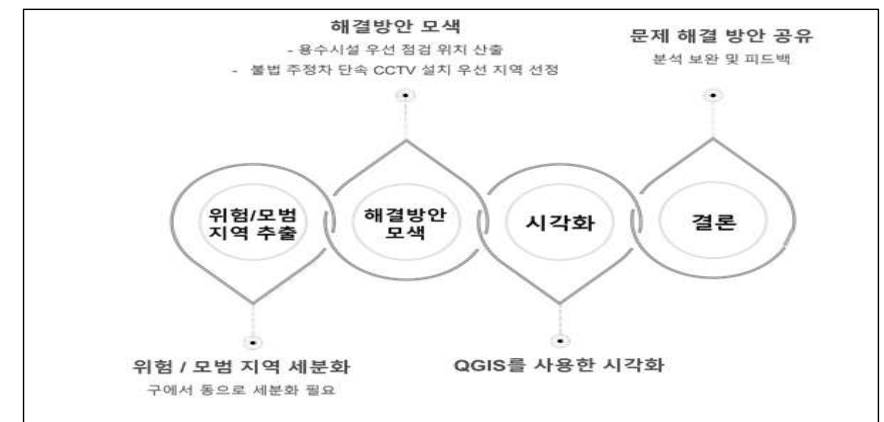
## 4.1. 분석 프로세스

- ☐ 파이썬과 qgis를 순차적으로 사용하여 분석 진행
- ☐ 파이썬을 사용하여 화재 위험등급을 분류하는 과정은 다음과 같음



[그림 4-1] 화재 위험등급 분석 프로세스

- ☐ QGIS를 사용하여 불법 주정차 단속 CCTV 설치 지역을 선정하는 과정은 다음과 같음



[그림 4-2] 화재 위험 지역 시각화 프로세스

## 4.2. 등급 산출을 위한 모델링

### 4.2.1 분석 방법

☐ Random Forest Classification

```
In [27]: from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()

rf.fit(x_tr, y_tr)
rf_pred = rf.predict(x_te)

print('정확도: (0, 4)')
format(accuracy_score(y_te, rf_pred))

<ipython-input-27-53c47ae5d338>:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,) for example using.ravel().
  rf.fit(x_tr, y_tr)

정확도: 0.9084
```

[그림 4-3] 랜덤포레스트 예측

- Random Forest로 등급 예측을 했을 때, 정확도는 약 0.90으로 나옴

- XGBoost Classification

```
In [31]: import xgboost

xgbC = xgboost.XGBClassifier()

xgbC.fit(x_tr, y_tr)

xgbC_pred = xgbC.predict(x_te)

print(' 정확도: {0.4f}'.format(accuracy_score(y_te, xgbC_pred)))

C:\Users\user\anaconda3\lib\site-packages\xgboost\sklearn.py:1145: UserWarning: The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [nuc_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

C:\Users\user\anaconda3\lib\site-packages\xgboost\sklearn.py:63: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return (rangs, **kwargs)

[17:32:15] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.4.0\src\learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softmax' was changed from 'error' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
정확도: 0.9348
```

[그림 4-4] XGBoost 예측

- XGBoost로 등급 예측을 했을 때, 정확도는 약 0.93으로 나옴

- CatBoost Classification

```

In [35]: from catboost import CatBoostClassifier

cb = CatBoostClassifier(loss_function='MultiClass')

cb.fit(x_train, y_train)

cb_pred = cb.predict(x_test)

print(' 정확도 : (0.4) ', format(accuracy_score(y_test, cb_pred)))

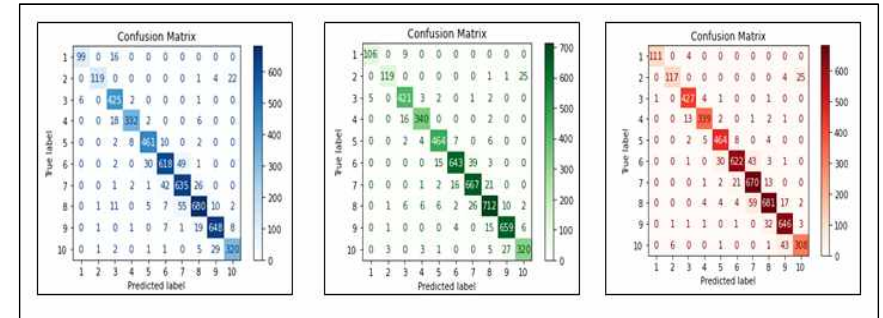
902: learn: 0.1306300 total: 22.1s remaining: 3.06s
903: learn: 0.1306549 total: 22.1s remaining: 3.06s
904: learn: 0.1306186 total: 22.1s remaining: 3.06s
905: learn: 0.1306696 total: 22.1s remaining: 3.13s
906: learn: 0.1305555 total: 22.1s remaining: 2.91s
907: learn: 0.1304664 total: 22.1s remaining: 2.68s
908: learn: 0.1303508 total: 22.1s remaining: 2.46s
909: learn: 0.1302381 total: 22.1s remaining: 2.24s
910: learn: 0.1301644 total: 22.2s remaining: 2.01s
911: learn: 0.1299629 total: 22.2s remaining: 1.73s
912: learn: 0.1298522 total: 22.2s remaining: 1.56s
913: learn: 0.1297560 total: 22.2s remaining: 1.34s
914: learn: 0.1296393 total: 22.2s remaining: 1.12s
915: learn: 0.1295722 total: 22.3s remaining: 89.4ms
916: learn: 0.1295395 total: 22.3s remaining: 67.1ms
917: learn: 0.1294780 total: 22.3s remaining: 44.7ms
918: learn: 0.1294211 total: 22.3s remaining: 22.3ms
919: learn: 0.1293622 total: 22.3s remaining: 0ms
정확도 : 0.9197

```

그림 4-5 CatBoost 예측

- CatBoost로 등급 예측을 했을 때, 정확도는 약 0.91으로 나옴

□ RandomForest/XGBoost/CatBoost 결과 혼돈행렬(confusion matrix)



[그림 4-6] Confusion Matrix 시각화

#### 4.2.2 등급 산출 모델링 결과

RandomForest	XGBoost	CatBoost
0.9084	0.9348	0.9197

[그림 4-7] 세 가지 알고리즘 예측 정확도 비교

- XGBoost가 가장 높은 예측률을 가짐

□ XGBoost를 사용하여 예측한 등급

시도	시군구	지연 종	소방서 거리 (km)	안전 센터 거리	다음 이동 업무 부	평화관 리대상 여부	자동차 재감지 기	자동차 소회 설비(작동 및과장)	소화기구 (사용·미 사용여부)	옥내소화 전(사용·미 사용여부)	건물중 수(지 상)	건물 중수 (지하)	배약연 격	유도 등	비상 경보 설비	비상 방송 설비	건물 구조조 제	장 소	통 과
서울	강남 시	노원 1동	4.1	3.0	N	Y	열감지 기	미설(미상)	미설	미설	6.0	1.0	155.00	미 설	미 설	미 설	철근 콘크리트 구조	화재 취약	2
서울	종로 시	서교 동	1.0	2.0	N	N	미설	미설(미상)	미설	미설	2.0	1.0	80.70	미 설	미 설	미 설	지상 벽돌조	화재 취약	5
서울	강서 시	가양 제1 동	2.1	3.7	N	Y	미설	미설(미상)	미설	미설	3.0	1.0	499.73	미 설	미 설	미 설	철근 콘크리트 구조	화재 취약	8
서울	강동 시	상도 제2 동	1.0	1.0	N	Y	미설	미설(미상)	사용	사용	12.0	1.0	815.78	미 설	미 설	미 설	철근 콘크리트 구조	화재 취약	7
서울	동대문 시	합천 동	2.7	1.3	N	Y	미설	미설(미상)	미설	미설	15.0	1.0	526.65	미 설	미 설	미 설	철근 콘크리트 구조	화재 취약	8

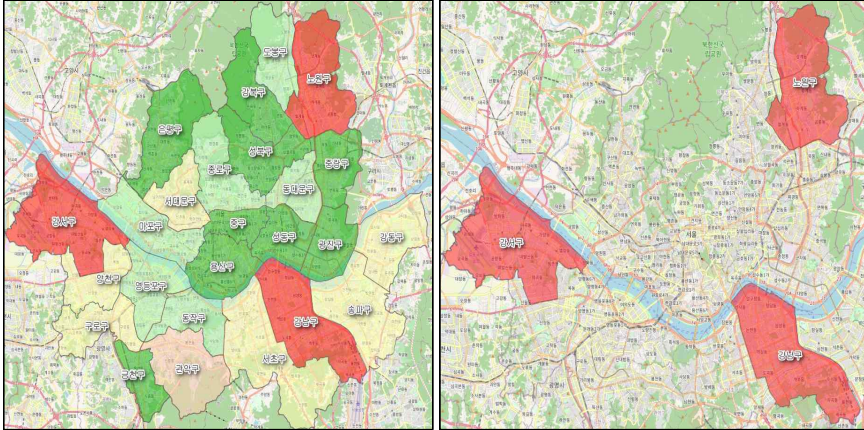
[그림 4-8] XGBoost 예측 등급



### 4.3. CCTV 설치 지역 선정 및 소방용수시설 점검 지역 선정 시각화

#### 4.3.1 위험등급 산출결과 시각화

- XGBoost 예측 등급을 활용하여 서울특별시 내 화재위험 지역구를 찾음  
- 위험지수 TOP 3 지역구 : 강서구, 강남구, 노원구

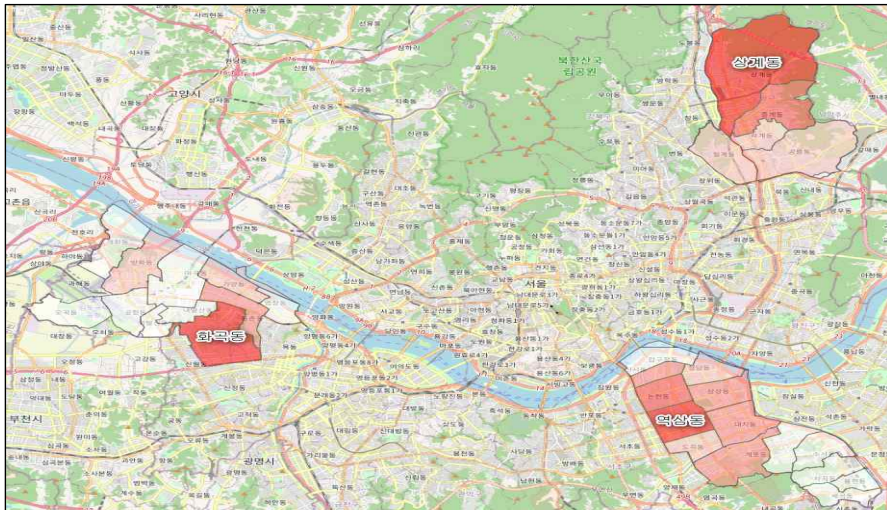


[그림 4-9] 서울특별시 화재 고위험 구 TOP 3

#### 4.3.2 화재 위험 TOP 3 지역의 Target 동 산출

- 해당 구에서 위험 등급이 가장 높은 동 추출

강서구 - 화곡동 / 강남구 - 역삼동 / 노원구 - 상계동



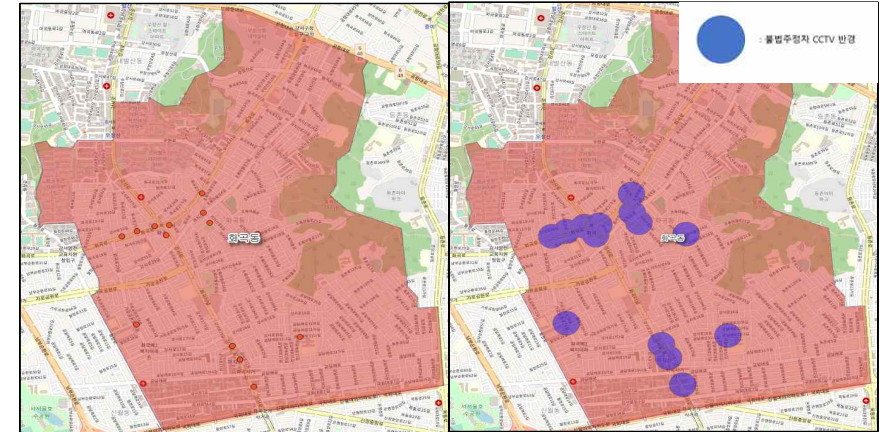
[그림 4-10] 서울특별시 화재 고위험 구 TOP 3

#### 4.3.3 지역별 불법 주정차 단속 CCTV와 용수시설 분석

- 고위험 동의 불법 주정차 단속 CCTV와 용수시설에 대한 QGIS 분석

##### 1) 강서구 - 화곡동 CCTV 설치 현황

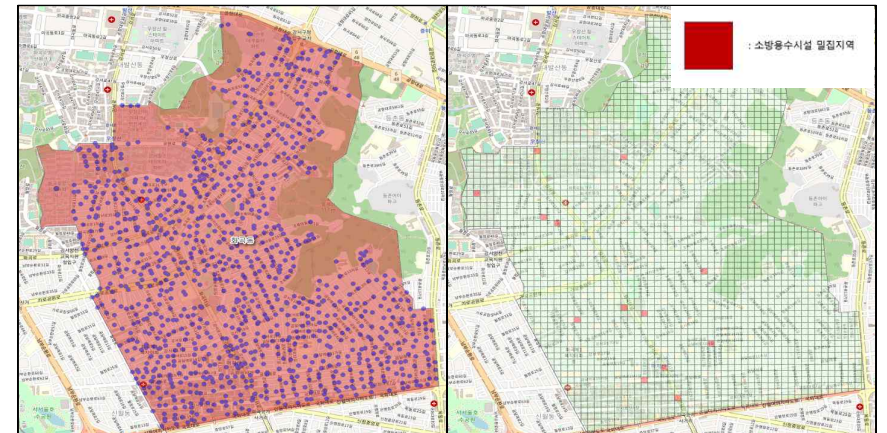
- CCTV 촬영 반경(통상 60~150m, 분석에서는 100m 적용)



[그림 4-11] 화곡동 CCTV설치 현황

##### 2) 강서구 - 화곡동 용수시설 위치 현황(소화전)

- 해당지역 그리드 생성(50m) 후 그리드 내 소방용수시설 밀집지역 산출



[그림 4-11] 화곡동 소방 용수시설 설치 현황

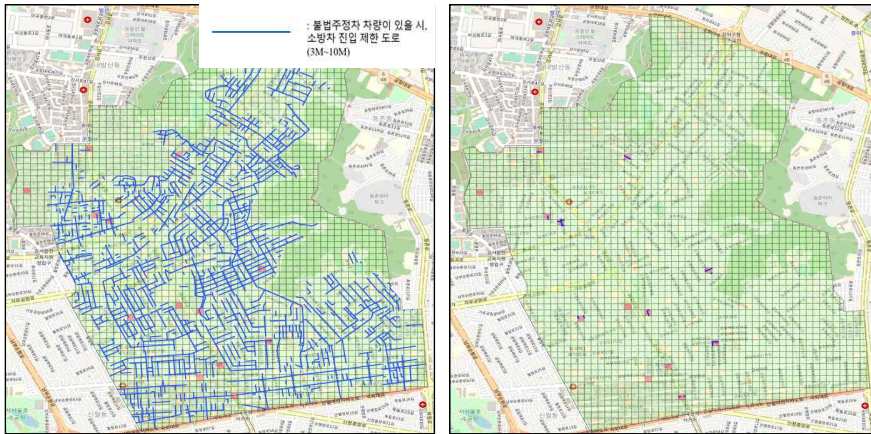


## 4.3.4 불법 주정차 CCTV 설치입지 시각화

- 해당 지역별 도로 데이터 사용 (불법 주정차 존재 시 소방차 진입 불가 도로(1) ) 및 최소 도로 폭 미확보로 소방차 진입 자체가 불가능한 도로(2))
- 도로 1 : 도로폭 3m 이상 ~ 10m 이하
  - 도로 2 : 도로폭 3m 미만

## 1) 불법 주정차로 인한 소방차 진입 제한 구역 시각화

- 소방용수시설 밀집 지역과 소방차 진입 제한도로가 겹치는 지점 산출



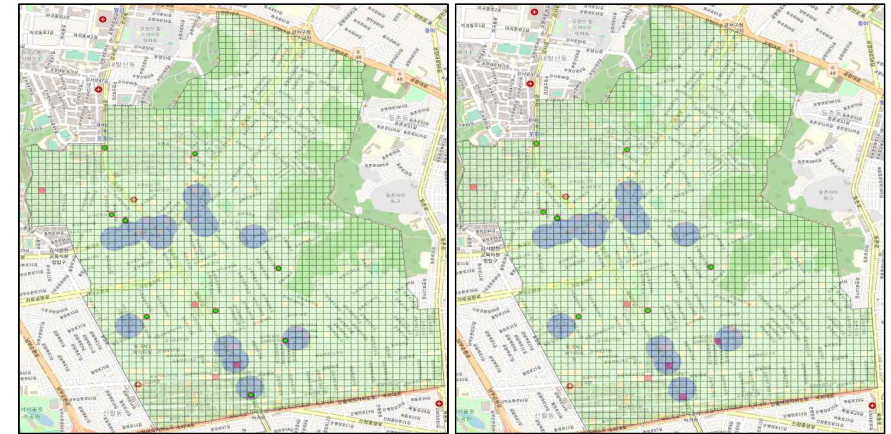
[그림 4-12] 화곡동 소방 용수시설 및 소방차 진입 제한도로 산출



[그림 4-13] 화곡동 CCTV 설치 지역 산출 (1차)

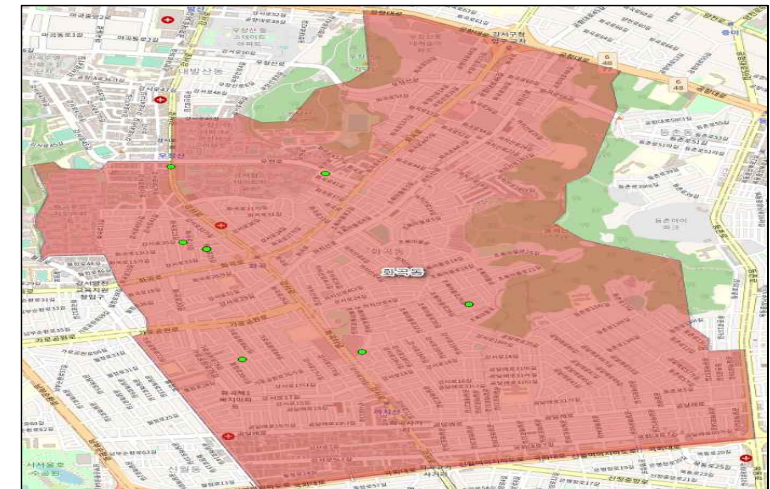
## 2) 최종 입지 선정

- 소방차 진입 제한 구역과 활동 중인 CCTV를 비교
- : 소방차 진입 제한 구역과 활동 CCTV 중복 확인(2개)



[그림 4-14] 화곡동 활동 중인 CCTV 시각화

- 총 7개의 CCTV 추가 설치 지역 산출

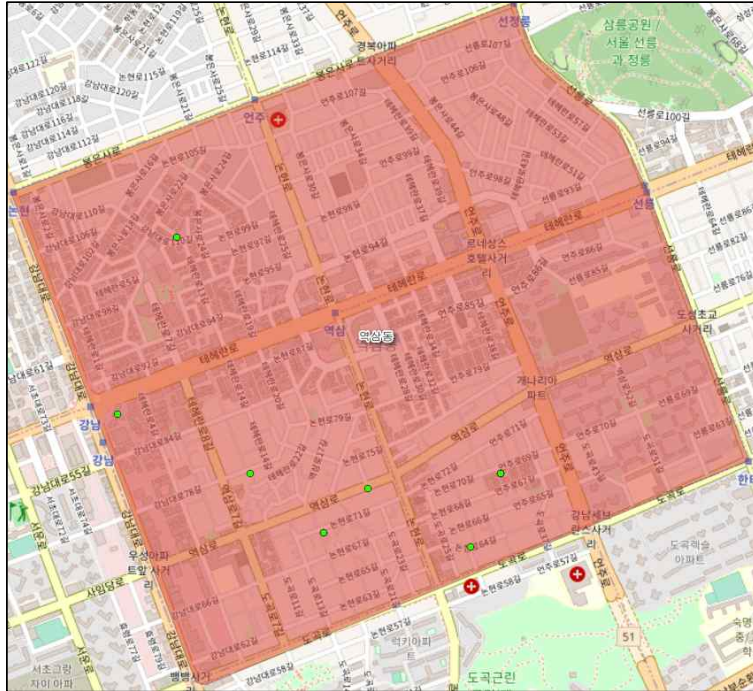


[그림 4-15] 화곡동 CCTV 설치 지역 최종 산출



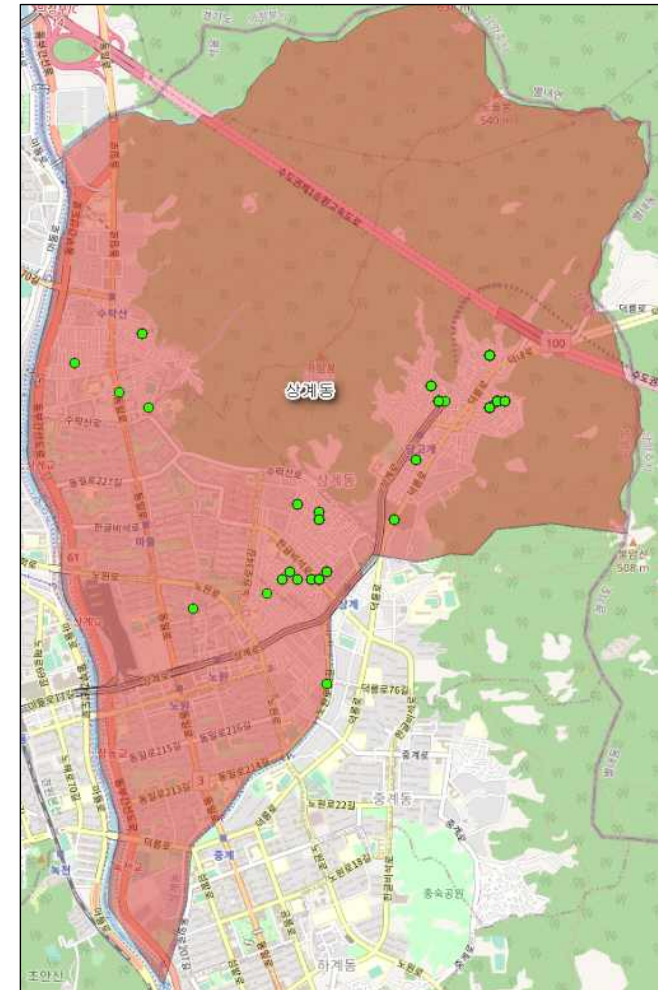
- 화곡동에서 최종 CCTV 설치위치 산출한 방법과 동일한 방법으로 상계동, 역삼동 시각화 결과

- 역삼동 : 총 7개의 CCTV 추가 설치 지역 선정



[그림 4-16] 역삼동 CCTV 설치 지역 최종 산출

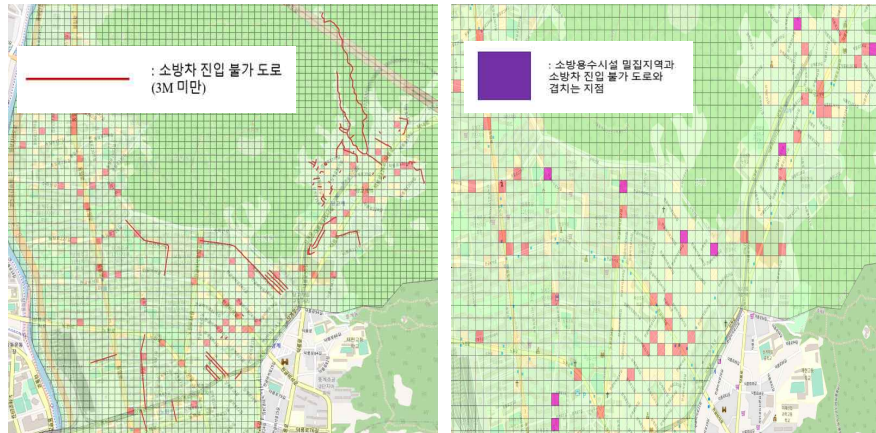
- 상계동 : 총 25개의 CCTV 추가 설치 지역 선정



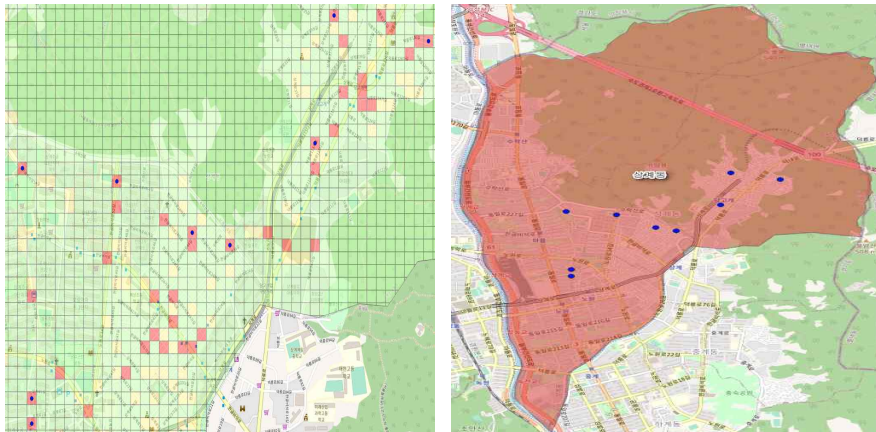
[그림 4-17] 상계동 CCTV 설치 지역 최종 산출

## 4.3.5 소방용수시설 밀집 지역 우선점검지점 선정(소방차 진입 불가 도로)

- 도로 폭 3m 미만으로 소방차 진입 불가도로에 대한 해결방안 시각화(역삼동, 화곡동은 해당 도로 내 소방용수시설 밀집 지역이 없어 상계동 한정)
- 상계동 내 도로 2 시각화(3m 미만 도로)
- 소방용수시설 밀집지역(빨간 그리드)와 3m 미만 도로 겹치는 지점 산출(보라색)



[그림 4-18] 상계동 소방차 진입 불가 지점 시각화



[그림 4-19] 상계동 소방용수시설 점검 입지 선정(최종)

- 총 9개의 소방용수 점검 지역 산출

## 5. 활용 방안

## 5.1. 업무 활용 방안

- 지역별 최적화된 소방 안전 관리 시스템 구축 및 대형 화재 취약 지역을 정밀 점검 대상으로 선정 및 관리
- 시구 및 행정동 별 화재 위험 등급 지도, 소방용수시설 현황 지도 제작 및 전략적 화재 대응 전략 수립
- 불법 주정차 단속 CCTV 현황과 소방용수시설 설치 현황을 파악하고, 지역별 화재 취약성을 고려한 불법 주정차 단속 CCTV 확충 대상 지역을 선별하여 소방용수시설 인근의 불법 주정차를 제재할 대안을 마련
- 소방차 통행 불가 구역(도로 폭 3M 미만)을 파악하여 해당 지역은 소방차 진입이 불가하니 소방용수시설 우선 점검 지역 선정하여 상시 사용 가능한 상태를 유지할 필요가 있음
- 소방차 통행제한구역(도로폭 3M 미만) 내에 불법 주정차 단속 CCTV를 입지 분석하여 소방용수시설 밀집 지역에 우선적 설치 구역을 지정

## 5.2. 문제점 개선 방안

## 1) 문제점

- 공공데이터 포털에 공개된 화재 발생 주소 정보는 2017년도에 한정됨
- 건물별로 위험 등급 부여 시 더 정확한 모델링이 가능하나, 각 건물별 등급 공개는 개인 정보법에 위반됨

## 2) 개선 방안

- 개인 정보를 보호할 수 있으면서 적절하게 사용 가능한 범위의 정보 공개 필요
- 공공기관(소방청)과 협력을 통해 지역별 위험 등급 모델을 확장하여 건물별 위험 등급 모델로 구체화하고 화재 취약 지역 예측 및 예방 가능
- 시간 제약으로 서울특별시로 한정하였으나, 공개된 화재 발생 주소 정보에 전국 데이터가 있어서 다른 지역에도 화재 위험 등급 모델 적용 가능





## 7.2 분석 상세코드

```
[1]: import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler

In [2]: df = pd.read_csv('fire_covov.csv', encoding = "EUC-KR")
#gu = pd.read_csv('gu_data.csv', encoding = "EUC-KR")
#dong = pd.read_csv('dong_data.csv', encoding = "EUC-KR")

C:\Users\user\anaconda3\lib\site-packages\PythonCore\InteractiveShell.py:3165: DtypeWarning: Columns (112,119,126,155,172) have mixed
types.Specify dtype option on import or set low_memory=False
has_raise = await self.run_ast_nodes(code_ast.body, cell_name,

In [3]: feature1 = ["시도", "시군구", "지번동", "화재유형", "소방서거리(km)", "안전센터거리", "다중이용업여부", "발화관리대상여부", "자동차재감지기",
feature2 = ["시도", "시군구", "지번동", "소방서거리(km)", "안전센터거리", "다중이용업여부", "발화관리대상여부", "자동차재감지기", "자동차보험"]

new_df = df[feature1]
df2 = new_df[['new_df['화재유형'] == '건축, 구조물']]
new_df2 = df2[feature2]
new_df2 = new_df2.reset_index()

In [4]: new_df2["자동차재감지기"] = new_df2["자동차재감지기"].fillna("미상")
new_df2["자동차보험비(작동및 효과성)"] = new_df2["자동차보험비(작동및 효과성)].fillna("미상(미상)"]
new_df2["소화기구(사용/미사용여부)"] = new_df2["소화기구(사용/미사용여부)"].fillna("미상")
new_df2["국내소화전(사용/미사용여부)"] = new_df2["국내소화전(사용/미사용여부)"].fillna("미상")
new_df2["건물층수(지상)"] = new_df2["건물층수(지상)"].fillna(1)
new_df2["건물층수(지하)"] = new_df2["건물층수(지하)"].fillna(0)
new_df2["유도등"] = new_df2["유도등"].fillna("미상")
new_df2["비상경보설비"] = new_df2["비상경보설비"].fillna("미상")
new_df2["비상방송설비"] = new_df2["비상방송설비"].fillna("미상")

In [5]: new_df2.to_csv('raw_data.csv', encoding = "EUC-KR")
raw = pd.read_csv('raw_data.csv', encoding = "EUC-KR")

In [8]: df_merge2 = new_df2

df_merge2["다중이용업여부"] = df_merge2["다중이용업여부"].replace("N", 0).replace("Y", 1)

df_merge2["자동차보험비(작동및 효과성)"] = df_merge2["자동차보험비(작동및 효과성)"].replace("효과적 작동(효과적작동)", 0)
df_merge2["자동차보험비(작동및 효과성)"] = np.where(df_merge2["자동차보험비(작동및 효과성)"] == 0, 0, 1)

df_merge2["국내소화전(사용/미사용여부)"] = df_merge2["국내소화전(사용/미사용여부)"].replace("사용", 0).replace("기타 국내소화전장동", 0)
df_merge2["국내소화전(사용/미사용여부)"] = np.where(df_merge2["국내소화전(사용/미사용여부)"] == 0, 0, 1)

df_merge2["비상경보설비"] = df_merge2["비상경보설비"].replace("경보", 0).replace("기타", 0)
df_merge2["비상경보설비"] = np.where(df_merge2["비상경보설비"] == 0, 0, 1)

df_merge2["비상방송설비"] = df_merge2["비상방송설비"].replace("방송", 0).replace("기타", 0)
df_merge2["비상방송설비"] = np.where(df_merge2["비상방송설비"] == 0, 0, 1)

df_merge2["자동차재감지기"] = df_merge2["자동차재감지기"].fillna(1)
df_merge2["자동차재감지기"] = df_merge2["자동차재감지기"].replace("미상", 1)
df_merge2["자동차재감지기"] = np.where(df_merge2["자동차재감지기"] == 1, 1, 0)

df_merge2["발화관리대상여부"] = df_merge2["발화관리대상여부"].replace("N", 1).replace("Y", 0)

df_merge2["소화기구(사용/미사용여부)"] = df_merge2["소화기구(사용/미사용여부)"].replace("사용", 0)
df_merge2["소화기구(사용/미사용여부)"] = np.where(df_merge2["소화기구(사용/미사용여부)"] == 0, 0, 1)

df_merge2["유도등"] = df_merge2["유도등"].replace("작동", 0).replace("기타", 0)
df_merge2["유도등"] = np.where(df_merge2["유도등"] == 0, 0, 1)

df_merge2["건물구조조"] = df_merge2["건물구조조"].replace("목조", 1).replace("비닐하우스 파이프조", 1).replace("샌드위치패널조", 1)
df_merge2["건물구조조"] = np.where(df_merge2["건물구조조"] == 1, 1, 0)

df_merge2["건물층수(지상)"] = df_merge2["건물층수(지상)"].fillna(1)
df_merge2["건물층수(지하)"] = df_merge2["건물층수(지하)"].fillna(0)
df_merge2["건물층수(지하)"] = df_merge2["건물층수(지하)"]+1.5

df_merge2["잠소문류"] = df_merge2["잠소문류"].replace("오탁시설", 1).replace("위락시설", 1).replace("일상서비스", 1)
df_merge2["잠소문류"] = np.where(df_merge2["잠소문류"] == 1, 1, 0)
```

```

In [10]: sc = MinMaxScaler()

feature3 = ["소방서거리(km)", "안전센터거리", "다중이용업부", "방화관리대상여부", "자동차재감지기",
            "자동차화재발생(작동일효과성)", "소화기구(사용/미사용여부)", "옥내소화전(사용/미사용여부)", "건물출수(지상)", "건물출수(지하)",
            "바닥면적", "유도등", "비상경보설비", "비상방송설비", "건물구조조", "질소중률류"]

new_df3 = df_merge2[feature3]

new_df3 = sc.fit_transform(new_df3)
new_df3 = pd.DataFrame(new_df3)
new_df3.columns = ["소방서거리", "안전센터거리", "다중이용업부", "방화관리대상여부", "자동차재감지기", "자동차화설비",
                  "소화기구", "옥내소화전", "건물출수(지상)", "건물출수(지하)", "바닥면적", "유도등", "비상경보설비", "비상방송설비",
                  "건물구조조", "질소중률류"]

In [12]: new_df3["결과"] = new_df3["소방서거리"]*0.0605 + new_df3["안전센터거리"]*0.0605
          + new_df3["다중이용업부"]*0.0605 + new_df3["방화관리대상여부"]*0.0605
          + new_df3["자동차재감지기"]*0.137 + new_df3["자동차화설비"]*0.0236 + new_df3["소화기구"]*0.0236
          + new_df3["옥내소화전"]*0.0236 + new_df3["건물출수(지상)"]*0.0686 + new_df3["건물출수(지하)"]*0.0686
          + new_df3["바닥면적"]*0.0686 + new_df3["유도등"]*0.055 + new_df3["비상경보설비"]*0.055
          + new_df3["비상방송설비"]*0.055 + new_df3["건물구조조"]*0.08 + new_df3["질소중률류"]*0.099

In [21]: result2["등급"] = ""

result2.loc[result2["결과"] <= 0.2, "등급"] = "1"
result2.loc[(result2["결과"] > 0.2) & (result2["결과"] <= 0.35), "등급"] = "2"
result2.loc[(result2["결과"] > 0.35) & (result2["결과"] <= 0.37), "등급"] = "3"
result2.loc[(result2["결과"] > 0.37) & (result2["결과"] <= 0.381), "등급"] = "4"
result2.loc[(result2["결과"] > 0.381) & (result2["결과"] <= 0.384), "등급"] = "5"
result2.loc[(result2["결과"] > 0.384) & (result2["결과"] <= 0.386), "등급"] = "6"
result2.loc[(result2["결과"] > 0.386) & (result2["결과"] <= 0.44), "등급"] = "7"
result2.loc[(result2["결과"] > 0.44) & (result2["결과"] <= 0.46), "등급"] = "8"
result2.loc[(result2["결과"] > 0.46) & (result2["결과"] <= 0.49), "등급"] = "9"
result2.loc[(result2["결과"] > 0.49, "등급"] = "10"

In [55]: seoul_raw["등급"] = pd.to_numeric(seoul_raw["등급"])
grp_result = seoul_raw.groupby(by=["시군구"], as_index=False).sum()
grp_result

grp_result.sort_values(by = "등급", ascending = False)

Out[55]:

```

```

In [36]: trte = rawl
trte_dummies = pd.get_dummies(trte, columns = ['다중이용여부', '방화관리대상여부', '자동화제감지기',
'자동소화설비(작동발효과설', '소화기구(사용/미사용여부)',
'옥내소화전(사용/미사용여부)', '유도등', '비상경보설비', '비상방송설비',
'건물구조조', '장소중분류'], drop_first = True)

tr = trte_dummies[trte_dummies["시도"] != "서울특별시"]
te = trte_dummies[trte_dummies["시도"] == "서울특별시"]

tr = tr.drop(["Unnamed: 0", "index", "시도", "시군구", "지번등"], axis = 1)
te = te.drop(["Unnamed: 0", "index", "시도", "시군구", "지번등"], axis = 1)

In [39]: x_tr = tr.drop(["등급"], axis = 1)
y_tr = tr["등급"]

x_te = te.drop(["등급"], axis = 1)
y_te = te["등급"]

y_tr = pd.DataFrame(y_tr)
y_tr.columns = ["등급"]

y_te = pd.DataFrame(y_te)
y_te.columns = ["등급"]

In [42]: from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix, mean_squared_error
from sklearn.ensemble import RandomForestClassifier

In [43]: rf = RandomForestClassifier()

rf.fit(x_tr, y_tr)
rf_pred = rf.predict(x_te)

print(' 정확도: {0:.4f}'.format(accuracy_score(y_te, rf_pred)))

<ipython-input-43-53c47ae5d38b>:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the
shape of y to (n_samples,), for example using ravel().
  rf.fit(x_tr, y_tr)
정확도: 0.9088

In [44]: confusion_matrix(y_te, rf_pred)
Out[44]: array([[100,  0, 15,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 120,  0,  0,  0,  0,  0,  1,  5, 20],
 [ 7,  0, 424,  3,  0,  0,  0,  0,  0,  0],
 [ 0,  0, 20, 331,  1,  1,  0,  5,  0,  0],
 [ 0,  0,  3,  6, 462, 11,  0,  1,  0,  0],
 [ 0,  0,  2,  0, 29, 621, 49,  1,  0,  0],
 [ 0,  0,  1,  1,  1,  47, 636, 21,  0,  0],
 [ 0,  0, 12,  2,  5,  7, 63, 670, 10, 21],
 [ 0,  2,  0,  1,  0,  5,  0, 23, 647, 7],
 [ 0,  2,  2,  0,  1,  1,  0,  6, 34, 313]], dtype=int64)

In [45]: import xgboost

In [46]: xgbC = xgboost.XGBClassifier()

xgbC.fit(x_tr, y_tr)
xgbC_pred = xgbC.predict(x_te)

print(' 정확도: {0:.4f}'.format(accuracy_score(y_te, xgbC_pred)))

C:\Users\User\anaconda3\lib\site-packages\xgboost\sklearn.py:1146: UserWarning: The use of label encoder in XGBClassifier is deprecated
and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when construc
ting XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)
C:\Users\User\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was passed when a 1d
array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return f(*args, **kwargs)

[17:20:47] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_1.4.0\src\learner.cc:1095: Starting in XGBoost 1.3.0, the de
fault evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to 'mlogloss'. Explicitly set eval_metric if
you'd like to restore the old behavior.
정확도: 0.9355

In [47]: confusion_matrix(y_te, xgbC_pred)
Out[47]: array([[106,  0,  9,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 119,  0,  0,  0,  0,  1,  1, 25],
 [ 5,  0, 421,  3,  2,  0,  1,  2,  0],
 [ 0,  0, 16, 340,  0,  0,  0,  2,  0,  0],
 [ 0,  0,  2,  4, 464,  7,  0,  6,  0,  0],
 [ 0,  0,  0,  0, 15, 643, 39,  3,  0,  0],
 [ 0,  0,  0,  1,  2, 16, 667, 21,  0,  0],
 [ 0,  1,  6,  6,  6,  2, 26, 712, 10, 21],
 [ 0,  1,  0,  0,  0,  4,  0, 15, 659,  6],
 [ 0,  3,  0,  3,  1,  0,  0,  5, 27, 320]], dtype=int64)

```

```

In [52]: from catboost import CatBoostClassifier

cb = CatBoostClassifier(loss_function='MultiClass')

cb.fit(x_tr, y_tr)

cb_pred = cb.predict(x_te)

print(' 정확도: {0:.4f}'.format(accuracy_score(y_te, cb_pred)))

392: learn: 0.1293175 total: 37.5s remaining: 64ms
982: learn: 0.1293175 total: 37.5s remaining: 64ms
983: learn: 0.1293064 total: 37.5s remaining: 61ms
984: learn: 0.1291928 total: 37.6s remaining: 57ms
985: learn: 0.1291077 total: 37.6s remaining: 53ms
986: learn: 0.1290280 total: 37.7s remaining: 49ms
987: learn: 0.1290174 total: 37.7s remaining: 45ms
988: learn: 0.1287855 total: 37.7s remaining: 42ms
989: learn: 0.1287066 total: 37.8s remaining: 38ms
990: learn: 0.1286660 total: 37.8s remaining: 34ms
991: learn: 0.1286417 total: 37.8s remaining: 30ms
992: learn: 0.1286038 total: 37.9s remaining: 26ms
993: learn: 0.1284864 total: 37.9s remaining: 22ms
994: learn: 0.1282607 total: 37.9s remaining: 19ms
995: learn: 0.1281451 total: 38s remaining: 15ms
996: learn: 0.1280891 total: 38s remaining: 11ms
997: learn: 0.1279095 total: 38s remaining: 7.2ms
998: learn: 0.1277993 total: 38.1s remaining: 3.1ms
999: learn: 0.1276550 total: 38.1s remaining: 0s

정확도: 0.9216

In [56]: confusion_matrix(y_te, cb_pred)
Out[56]: array([[111,  0,  4,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 117,  0,  0,  0,  0,  0,  0,  4, 25],
 [ 1,  0, 427,  4,  1,  0,  0,  1,  0,  0],
 [ 0,  0, 13, 339,  2,  0,  1,  2,  1,  0],
 [ 0,  0,  2,  5, 464,  8,  0,  4,  0,  0],
 [ 0,  0,  1,  0, 30, 622, 43,  3,  1,  0],
 [ 0,  0,  0,  1,  2, 21, 670, 13,  0,  0],
 [ 0,  0,  0,  4,  4,  4, 59, 681, 17, 21],
 [ 0,  1,  1,  1,  0,  1,  0, 32, 646,  3],
 [ 0,  6,  0,  0,  1,  0,  0,  1, 43, 308]], dtype=int64)

```