2020년 공공 빅데이터 청년 인턴십 확대 운영 데이터 전문교육과정

데이터기반 행정으로 국민의 삶의 질을 개선하라!

데이터 인턴십 해커톤

분석결과 보고서

분석 주제명

전기 오토바이 교체소 입지 선정

참여자

신홍조, 강재구, 안태훈

박혜진, 손준형, 오종민

윤승후, 김병우, 배동준

씨에스리 컨소시엄

CSLEE kpc 한국생산성본부

목 차

1. 분석 개요

가. 분석 배경 및 개요

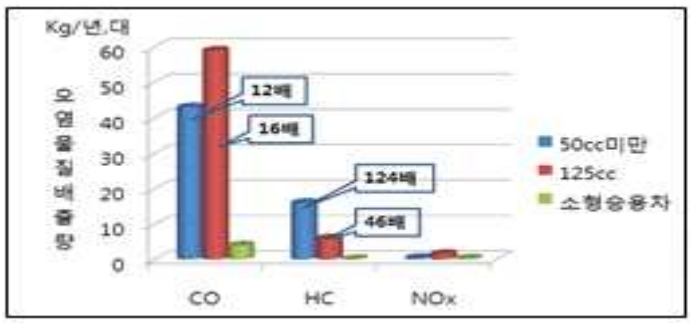
2020. 06. 05. 광화문 1번가에 한 민원이 올라왔다. 코로나로 인하여 배달이 많아져, 배달부들이 오토바이 공회전을 하며 공기 질을 오염시키고 소음을 일으킨다는 내용이다. 이에 배달 오토바이의 전기 오토바이 의무화 문제가 화두가 되었다

.

온라인 제안 **배달오토바이의 전기오토바이 의무화**

피자, 배달삼겹살 등 각종 음식업소들이
배달대행 오토바이 등으로 배달 위주로 영업을 하고 있습니다.
그리고 업소 오토바이도 따로 있어서 배달을 합니다.
문제는 이런 업소들이 빌라 등의 주택 바로 옆에 영업허가 되어
주민들이 수시로 드나드는 배달 오토바이 소음에 밤낮없이 시달린다는 것입니다.
더하여 시동을 켜둔 채로(공회전) 5분 10분 15분 이상을 음식업소에서
정차하여 두는 경우가 자주 있어서 소음, 매연, 먼지 등은 더욱더 가중됩니다.
주택가에 배달 오토바이들이 점점 더 활보하며 다니는 추세이기도 합니다.
그리하여, 배달 오토바이는 전기 오토바이로 의무화 하는 것이 시급하다는 생각입니다.
전기 오토바이로 교체되면 소음, 매연, 먼지 공해는 자연히 해결되리라 봅니다.
최근에 한국형 뉴딜정책의 하나로 경유차를 전기차로 교체할 계획에 있다는 뉴스를 보았습니다.
전기 오토바이 보급을 위하여 배터리, 충전거리문제 등 관련 기술 및 인프라
보조금지원, 홍보에 힘써서 배달 오토바이의 전기 오토바이 의무화 법안 마련이
하루 빨리 현실화 되기를 바랍니다.

출처 : 서울특별시 광화문 민원



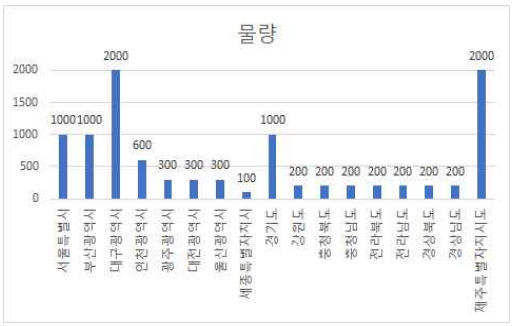
우리나라에서 실제로 오토바이가 내뿜는 매연의 실태는 심각하다. 엔진 승용차에 비해 125cc이하가 내뿜는 오염물질의 양은 최대 124배에 달한다.

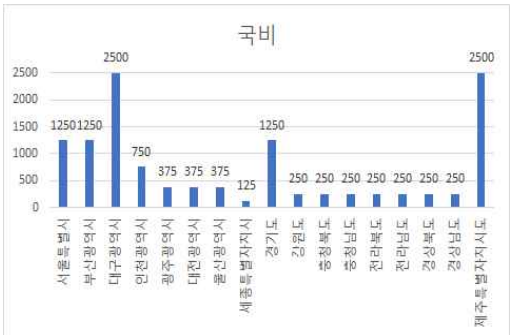
늘어난 탄소 배출량을 줄이는 것은 세계가 직면한 과제이며 우리나라 역시 예외는 아니다.

국가교통부는 이에 맞추어 그린뉴딜 정책을 발표 하였다. 이 중 이륜차 부문에서 그린배달 서포터즈를 출범해 배달용 오토바이를 전기 오토바이로 전환하고, 충전인프라 확충과 배터리 성능 개선을 적극 추진한다는 계획을 발표했다.

<정부 친환경 그린 모빌리티 보급 계획>

구분	물량	총 예산	국비	지방비
총 계	10,000	25,000	12,500	12,500
서울특별시	1,000	2,500	1,250	1,250
부산광역시	1,000	2,500	1,250	1,250
대구광역시	2,000	5,000	2,500	2,500
인천광역시	600	1,500	750	750
광주광역시	300	750	375	375
대전광역시	300	750	375	375
울산광역시	300	750	375	375
세종특별자치시	100	250	125	125
경기도	1,000	2,500	1,250	1,250
강원도	200	500	250	250
충청북도	200	500	250	250
충청남도	200	500	250	250
전라북도	200	500	250	250
전라남도	200	500	250	250
경상북도	200	500	250	250
경상남도	200	500	250	250
제주특별자치도	2,000	5,000	2,500	2,500

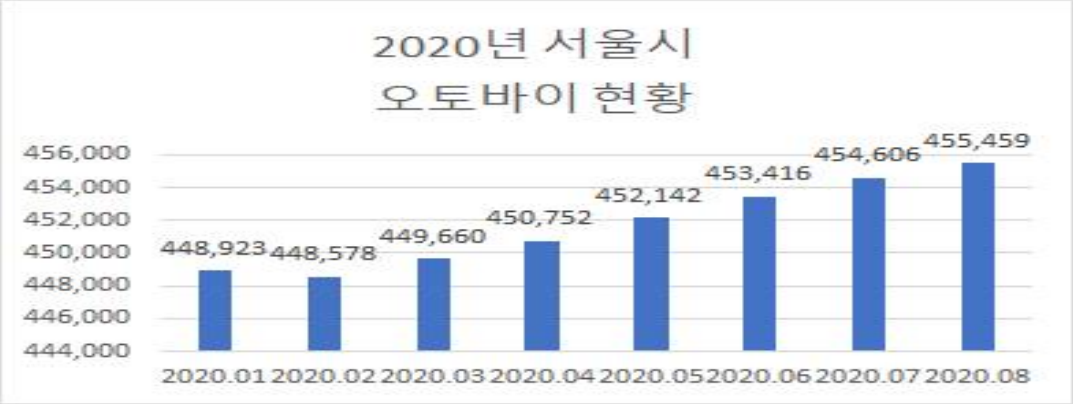


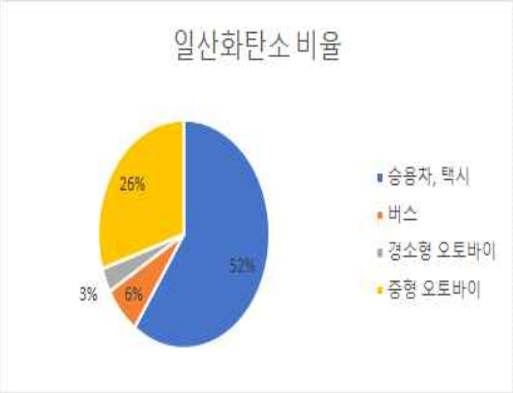


한국판 뉴딜 정책은 코로나19로 인한 경제침체를 극복하기 위한 프로젝트이다. 뉴딜정책은 1929년부터 발생한 미국의 경제 대공황으로 이것을 극복하기 위해 추진했던 정책으로, 정부는 이를 제반 삼아 한국 맞춤형 뉴딜 정책을 기획했다.

<한국형 뉴딜 사업>

한국판 뉴딜정책은 크게 디지털뉴딜과 그린뉴딜이 있다. 이 중 그린 뉴딜정책의 일환인 전기 오토바이 보급을 달성하기 위하여, 전기 오토바이의 치명적인 단점인 배터리 문제점을 해결해야 한다. 이 문제점을 해결하기 위하여, 9조에서는 빅데이터 분석을 활용한 배터리 교체소의 최적 입지 선정 기획을 진행하였다.





서울시 오토바이가 매월 증가하는 추세이다. 내연기관 오토바이의 환경오염물질 배출 및 소음 등으로 인해 쾌적한 주거환경 또한 침해 받고 있다. 이러한 문제점을 해소하기 위해 배출가스와 소음을 원천적으로 차단하는 전기 오토바이의 보급을 공공 부문에서 민간 부분으로까지 확대 실시해야 한다. 내연기관 오토바이 1대를 전기 오토바이로 교체하면 이산화탄소 발생량을 기준으로 연간 62그루의 소나무를 심는 대체효과를 얻을 수 있기 때문에 환경적 대책에 박차를 가해야 한다.

<친환경 전기 오토바이 편익>

구분	평균연비	연료가격	km당 연료비	연간연료비(연간 12천km 주행기준)
공공기관(대)	40.9km/kWh	56.2원/kWh	1.37원	16,400원/년
재원	33km/L	1,880원/L	56.97원	683,640원/년

* CO, NOx, THC, PM10 배출량 35kg/년·대 저감 : 550,000원/년
* CO2 환경편익 669kg/년·대 저감 : 26,000원/년

나. 분석목적 및 방향

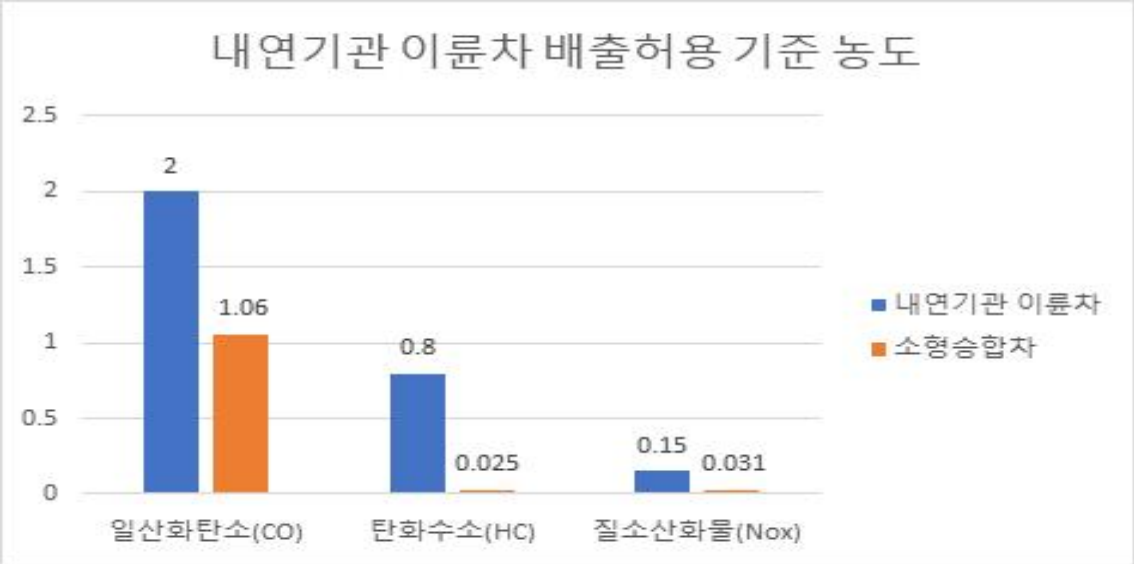
<베트남 오토바이 현황>

오토바이를 많이 사용하는 베트남에서는 4300 만대의 오토바이가 등록되어 있다. (자동차 200 만대)의 20 배 이상)
호치민시 자원환경국 환경관측분석센터 보고서에 따르면 휘발유를 배출하는 내연기관 오토바이에서 발생되는 배기가스가 베트남 도시 공기오염원의 70%를 차지한다. 시내 대기 중 일산화탄소(CO)와 부유 입자상 물질(SPM) 농도도 기준치를 크게 초과했다.

오염된 공기로 인한 폐질환도 심각하다. 베트남 풀브라이트대(Fullbright University)가 내놓은 연구보고서에 따르면 베트남에는 150만 명이상이 만성 폐쇄성 폐 질환을 앓고 있다. 2013 년 한해 4만 명이 같은 질환으로 사망했다. 교통사고 사망률보다 4 배 많다.

<국내 내연기관 오토바이 배출허용기준>

구분	일산화탄소(CO)	탄화수소(HC)	질소산화물(NOx)
내연기관 이륜차	2.0g/Km	0.8/Km	0.15g/Km
소형승합차	1.06g/Km	0.025g/Km	0.031g/Km



<국내 내연기관 오토바이 배출허용기준>

이러한 내연기관 운송수단의 유해성 때문에, 우리 정부에서는 각종 배출 규제와 친환경 운송수단 보급을 장려하여 대기오염에 대응하고 있다. 또한 친환경 사업을 위한 정책으로 2017년부터 그린모빌리티에 보조금을 210만원~330 만원을 지원했지만 저조한 판매량을 보인 바 있다. 이는 전기 오토바이가 내연기관의 오토바이에 비해 효율성이 떨어지기 때문이다.

<그림 1. 국내 전기 오토바이 보급 현황>

<그림 1. 2020년 서울시 오토바이 등록현황 추이>

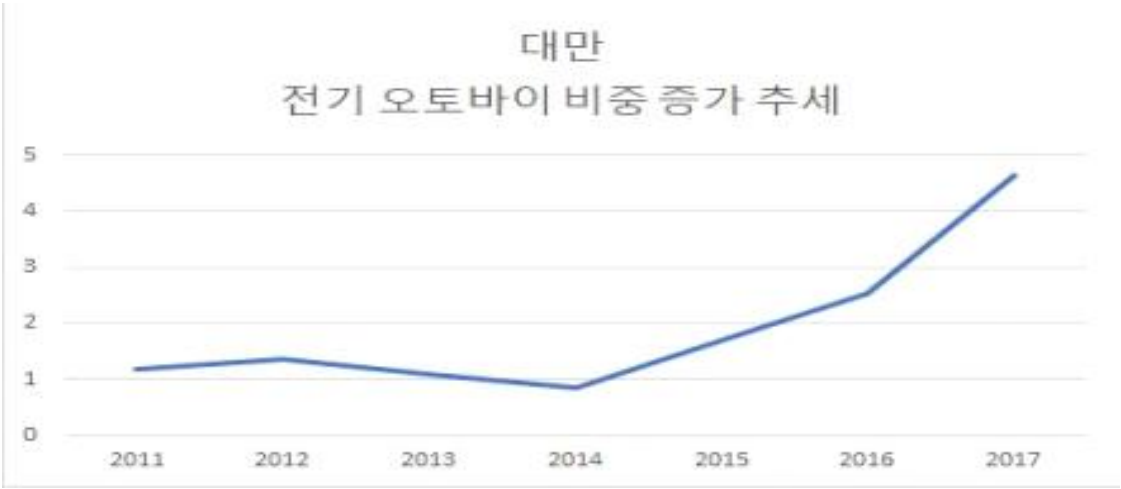
하지만 2018년을 기점으로 전기 오토바이 보급대수가 급증하는 모습을 보여준다. 이는 국내 중소기업이 개발한 오토바이의 가격이 보조금을 포함하여 60만 원 대로 형성되었고, 성능까지 개선되었기 때문이다. 개선된 전기 오토바이는 1회 완충 시 갈 수 있는 거리가 50km, 최대속도 80km/h로 출퇴근과 일상생활에 충분한 수준까지 발전하였다.

최근에는 음식 배달이 늘어나며 오토바이 수요도 증가하고 있다. 서울 서초구 소재 오토바이 매장 B 대표는 “배달용 중고 오토바이 문의를 하는 사람들은 부쩍 많아졌다” 말했고, 다른 관계자도 “무엇보다 고객들이 배달을 많이 시키고 있는 게 영향을 주고 있다”고 말했다. 실제 배달의 민족의 올 5월 주문 증가율은 전월 대비 63% 늘어났으며 4월 신규 배달 라이더는 1월 대비 208%폭 증했다.

하지만 오토바이 수요의 대부분을 차지하는 배달 업계에서는 전기 오토바이를 선호하지 않는다. 현재 시중의 오토바이는 배터리 충전에 3시간가량 소요되기 때문이다. 시간이 생명인 배달 업계 종사자에게는 내연기관보다 연료 보급이 오래 걸리는 전기 오토바이를 굳이 선택할 이유가 없다.
이 같은 문제점을 극복하기 위해, 대만에서는 전기 오토바이 교체형 배터리 방식과 배터리 교체소 인프라를 구축하여 충전시간을 획기적으로 단축하였고, 또한 전기 오토바이를 활성화하기 위한 다양한 정책들로 보급률을 끌어 올리는데 성공하였다.

<대만 전기차 보급량(단위 : 천대, %)>

	2011	2012	2013	2014	2015	2016	2017
전기 오토바이	7.6	8.5	7.2	5.6	12.0	21.6	46.7
내연 오토바이	639.6	616.3	660.7	660.5	695.2	831.5	955.5
비중	1.17	1.36	1.08	0.84	1.69	2.53	4.66



<대만 전기차 보급량>

이러한 선례를 바탕으로, 전기 오토바이의 성공적인 보급을 위해 효과적인 인프라 구축이 요구된다. 하지만 충전 인프라 구축비용은 정부의 계획된 재정 안에서만 가능하기 때문에 최적의 교체소를 선정하는 것이 필요하다.

다. 분석 결과 활용 방안

- 분석 결과를 바탕으로 오토바이 운행량이 많은 지역에 전기 오토바이 배터리 교체소 입지 우선 선정
- 분석 결과를 바탕으로 배터리 교체소 인프라를 점진적으로 확충하여 전기 오토바이의 보급 확산
- 분석 결과를 바탕으로 기존 배터리 충전소의 입지 타당성 재검토 가능
- 내연기관 오토바이로 인한 민원과 환경문제를 전기 오토바이를 통해 감축
- 전기 오토바이 실사용자의 만족도 향상

2. 분석 데이터

가. 분석 데이터 목록

<활용 데이터 목록>

구분	데이터명	데이터정의
공공개방	주민등록인구현황	- 서울특별시 각 지역구 동별 주민등록 인구
	인구주택 총 조사	- 서울특별시 각 지역구 동별 2018 1인 가구 변화 수
	통계청 전국 사업체 조사	- 서울특별시 각 지역구 동별 2018 사업체 등록 수 - 서울특별시 각 지역구 동별 2018 사업체 종사자수 - 서울특별시 각 지역구 동별 2018 도소매업 등록 수 - 서울특별시 각 지역구 동별 2018 서비스업 등록 수 - 서울특별시 각 지역구 2016~2018 동별 2018 치킨전문점 증가 수 - 서울특별시 각 지역구 2016~2018 동별 2018 커피전문점 증가 수 - 서울특별시 각 지역구 2016~2018 동별 2018 제과점 증가 수 - 서울특별시 각 지역구 2016~2018 동별 2018 PC방 증가 수 - 서울특별시 각 지역구 2016~2018 동별 슈퍼마켓 증가 수
	서울특별시 동별 경계	- 행정구역 시, 군구 경계
지자체	서울특별시 각 지역구 오토바이 등록 수	- 서울특별시 각 지역구 오토바이 등록 수 - 서울특별시 각 지역구 경형, 소형, 중형, 대형 오토바이 등록 수
	서울시 숙박 및 음식점업	- 서울특별시 각 지역구 동별 2018 숙박 및 음식점업 사업체 수 - 서울특별시 각 지역구 동별 2018 숙박 및 음식점업 종사자 수
민간데이터	gogoro station	- 서울시 內 gogoro station 설치 및 설치 예정 장소

<활용 데이터 제공처 및 형태>

구분	데이터 명	데이터 제공처	데이터 형태
공공개방	주민등록인구현황	행정안전부	CSV
	인구주택 총 조사	통계청	CSV
	통계청전국사업체조사	통계청	CSV
	서울시 동별 경계	통계청	JSON
지자체	서울특별시 각 지역구 오토바이 등록 수	서울 열린데이터 광장	CSV
	서울시 숙박 및 음식점업	서울 열린데이터 광장	CSV
민간데이터	gogoro station	티아이씨코퍼레이션	PNG

나. 데이터 상세 설명

<Y를 선정하기 위한 상관분석 변수 설명표>

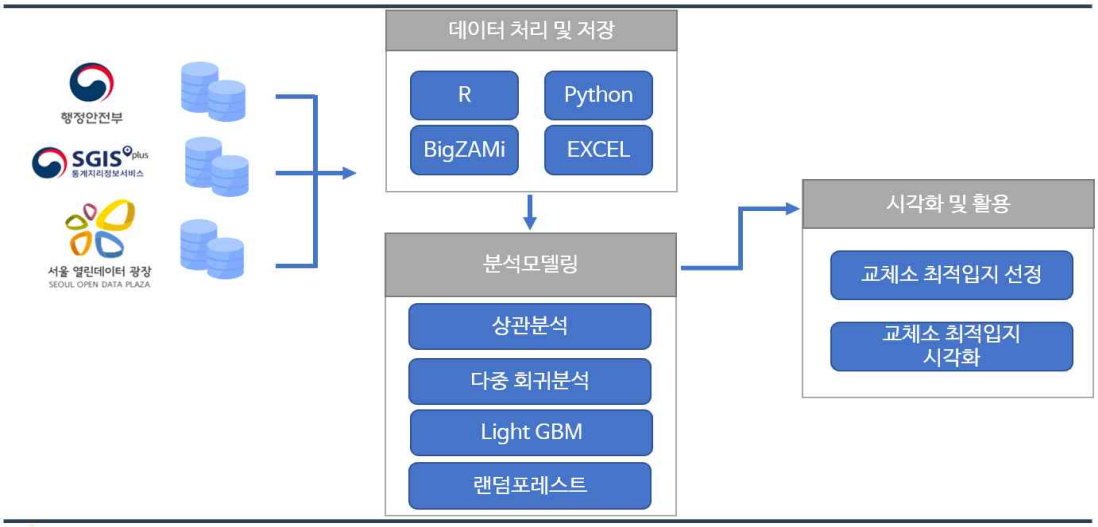
변수	변수설명	데이터 상세 설명	자료출처
X1_1	2010.01~2020.08 서울특별시 각 지역구 이륜차 등록 수	2010.01~2020.08 서울특별시에 등록되어 있는 이륜차 등록 수	서울열린데이터광장
X2_1	2020.01 서울특별시 각 지역구 이륜차 등록 수	2010.01 서울특별시에 등록된 각 지역구 이륜차 대수 합계	서울열린데이터광장
X3_1	2020.08 서울특별시 각 지역구 이륜차 등록 수	2020.08 서울특별시에 등록된 각 지역구 이륜차 대수 합계	서울열린데이터광장
X4_1	2020.08 서울특별시 각 지역구 경형 이륜차 등록 수	2020.08 서울특별시에 등록된 경형 이륜차 대수	서울열린데이터광장
X5_1	2020.08 서울특별시 각 지역구 소형 이륜차 등록 수	2020.08 서울특별시에 등록된 소형 이륜차	서울열린데이터광장
X6_1	2020.08 서울특별시 각 지역구 중형 이륜차 등록 수	2020.08 서울특별시에 등록된 중형 이륜차 대수	서울열린데이터광장
X7_1	2020.08 서울특별시 각 지역구 대형 이륜차 등록 수	2020.08 서울특별시에 등록된 대형 이륜차 대수	서울열린데이터광장
X8_1	2020.08 서울특별시 각 지역구 경형+소형 이륜차 등록 수	2020.08 서울특별시에 등록된 경형, 소형 이륜차 대수	서울열린데이터광장
X9_1	2020.08 서울특별시 각 지역구 경형+소형+중형 이륜차 등록 수	2020.08 서울특별시에 등록된 경, 소, 중형 이륜차 대수	서울열린데이터광장
Y1_1	서울특별시 각 지역구 2018 주민등록 인구	서울특별시 각 지역구 동별 주민등록인구	주민등록인구현황 (행정안전부)
Y2_1	서울특별시 각 지역구 2000~2018 인구 변화수	서울특별시 각 지역구 2000~2018년 주민등록인구 증가 수	서울특별시 인구주택총조사(통계청)
Y3_1	서울특별시 각 지역구 2018 1인 가구 변화 수	대한민국 모든 사람과 주택 조사	통계청전국 사업체 조사(통계청)
Y4_1	서울특별시 각 지역구 2018사업체 등록 수	대한민국 사업체 구조 파악을 위한 통계조사	통계청전국사업체조사 (통계청)
Y5_1	서울특별시 각 지역구 2018 1인당 각 치킨 인구 수	대한민국 사업체 구조 파악을 위한 통계조사	통계청전국사업체조사 (통계청)
Y6_1	서울특별시 각 지역구 2018 도소매업 등록 수	대한민국 사업체 구조 파악을 위한 통계조사	통계청전국사업체조사 (통계청)
Y7_1	서울특별시 각 지역구 2018 서비스업 등록 수	대한민국 사업체 구조 파악을 위한 통계조사	서울시 사업체 현황 통계
Y8_1	서울특별시 각 지역구 2018 음식점 등록수	대한민국 사업체 구조 파악을 위한 통계조사	서울시 사업체 현황 통계
Y9_1	서울특별시 각 지역구 2018 숙박 및 음식점업 사업체 수	대한민국 사업체 구조 파악을 위한 통계조사	통계청전국사업체 조사
Y10_1	서울특별시 각 지역구 2018 숙박 및 음식점업 종사자 수	대한민국 사업체 구조 파악을 위한 통계조사	통계청전국사업체조사

<다중회귀, 랜덤포레스트, Light GBM 변수 설명표>

변수	변수설명	데이터 상세 설명	자료출처
Y	서울특별시 각 지역구 동별 주민 등록인구	서울특별시 각 지역구 동별 주민등록 인구	주민등록인구현황 (행정안전부)
X1	서울특별시 각 지역구 동별 2018 1인 가구 변화 수	대한민국 모든 사람과 주택 조사	서울특별시 인구주택총조사 (통계청)
X2	서울특별시 각 지역구 동별 2018 사업체 등록 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X3	서울특별시 각 지역구 동별 2018 사업체 종사자 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X4	서울특별시 각 지역구 동별 2018 도소매업 등록 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X5	서울특별시 각 지역구 동별 2018 서비스업 등록 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X6	서울특별시 각 지역구 동별 2018 숙박 및 음식점 사업체 수	서울시에 등록된 종사자 1인 이상의 사업체 및 종사자	서울시 사업체현황 (서울열린데이터광장)
X7	서울특별시 각 지역구 동별 2018 숙박 및 음식점 종사자 수	서울시에 등록된 종사자 1인 이상의 사업체 및 종사자	서울시 사업체현황 (서울열린데이터광장)
X8	서울특별시 각 지역구 2016~2018 동별 2018 치킨전문점 증가 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X9	서울특별시 각 지역구 2016~2018 동별 2018 커피전문점 증가 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X10	서울특별시 각 지역구 2016~2018 동별 2018 제과점 증가 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X11	서울특별시 각 지역구 2016~2018 동별 2018 PC방 증가 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)
X12	서울특별시 각 지역구 2016~2018 동별 슈퍼마켓 증가 수	대한민국 사업체 구조 파악 하기 위한 전수 조사	통계청전국사업체조사 (통계청)

3. 분석 프로세스

가. 분석 프로세스



<분석 프로세스>

요건정의	모델링	검증 및 테스트	적용
종속변수와 독립변수의 독립변수를 설정한 뒤 종속변수를 추정	다중회귀분석	상관분석, 분산분석, 회귀모형 가정사항 확인(정규성, 등분산성, 독립성), VIF(다공선성)	예측한 회귀모형에 따른 결과 분석 후 결정계수에 따른 적합한 분석 여부 판단

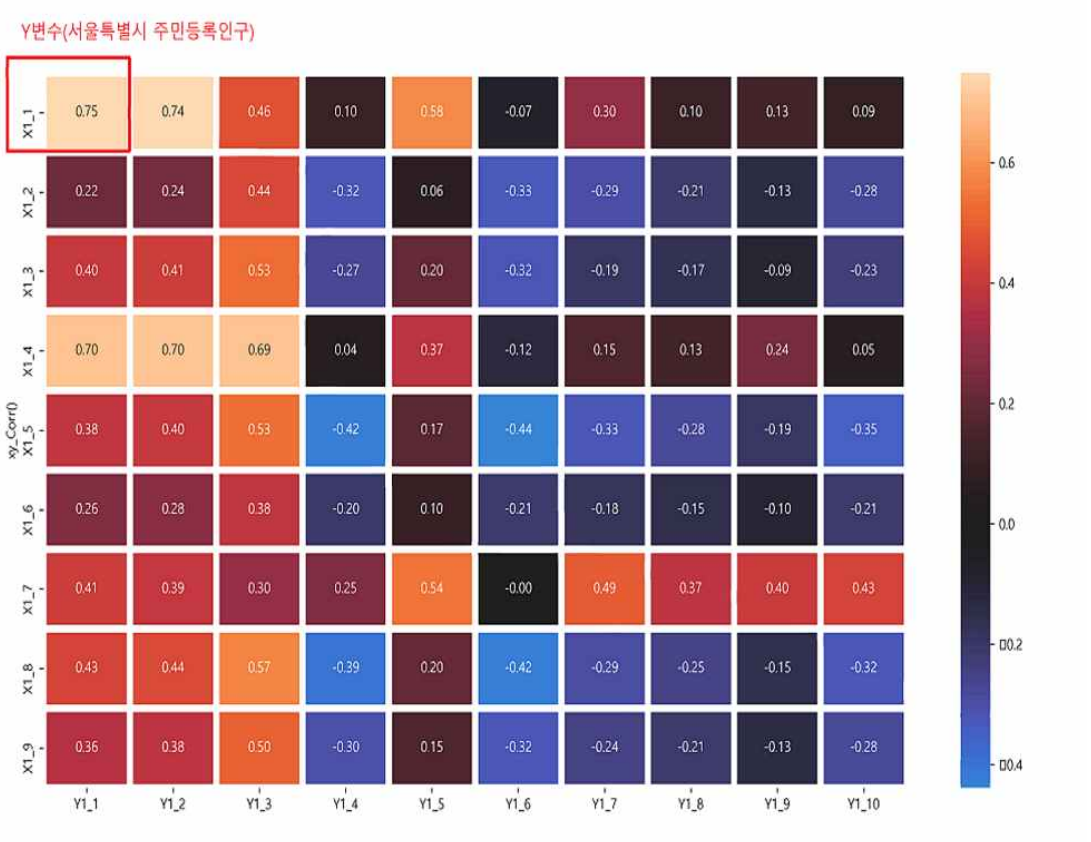
요건정의	모델링	검증 및 테스트	적용
데이터 전처리 및 모델 정확도 확인	Light GBM	최적모델생성 정확도평가 모델 성능 평가	랜덤포레스트 모델의 정확도, 성능과 비교

요건정의	모델링	검증 및 테스트	적용
종속변수와 독립변수 설정 변수 중요도 확인	랜덤포레스트	그리드 검색, 정확도, f1 score	서울시 내에 있는 각 동(읍면동)의 오토바이 수요정도(1~4단계)를 예측

전기 이륜차 교체소 최적의 입지선정을 위해 '주민등록 인구'와 많은 설명변수를 선정하고 "랜덤포레스트" 모델링을 통해 최적의 입지 선정을 선택하여 서울시 내에 있는 각 동(읍면동)의 오토바이 수요 정도(1~4단계)를 예측함.

나. 분석 내용 및 방법

상관분석



<Y변수 선정 상관분석>

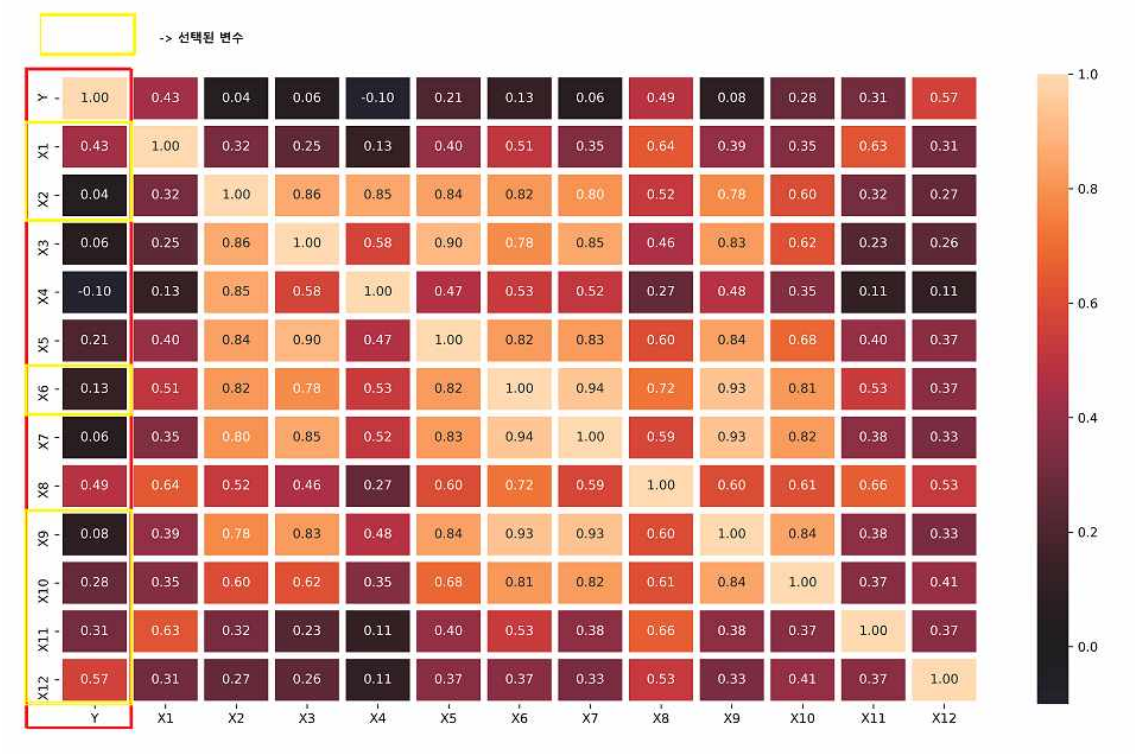
서울특별시의 동별 오토바이 보유량 자료가 존재하지 않아, 대체변수를 선정하기 위해 상관분석을 실시한다. 서울특별시 지역구 이륜차 등록데이터를 X로 선정한 뒤, 연관성이 있는 대체자료를 Y에 선정한다. 분석결과 X1_1(10년간 이륜차 등록 대수)와 Y1_1(서울특별시 주민등록인구)이 상관성이 높은 것으로 나타난다. 대부분이 무관성을 띄지만, 일부 상관성이 있다고 나오는 결과들보다 범용성이 뛰어난 Y1_1(서울특별시 주민등록인구)를 본 분석의 종속변수로 선정한다.

다중회귀분석

다중회귀분석은 하나의 종속변수와 하나의 독립변수로 분석하는 단순회귀분석에서 확장된 모델로 두 개 이상인 회귀모형을 다루는 것으로 분석의 단계로는 설정, 추정, 검정, 가정사항, 이용의 단계로 나뉜다. 본 분석은 오토바이 배터리 교체소로 적합한 입지를 선정하기 위해 앞의 분석의 결과인 서울특별시 지역구 각 동별 주민등록인구를 종속변수로(이하 Y) 사용한다.

설정

먼저, 설정의 단계는 추정하고자 하는 모형을 설정을 하고 난 뒤 데이터를 수집한다. 다음은 독립변수(Y)와 종속변수(X)의 산점도를 본 뒤 선형성을 파악하고 무관성을 띄는 변수는 제거하거나 변수변환을 한다. 종속변수(Y)와 독립변수(X)의 관계를 파악한 뒤, 독립변수들 간의 무관성을 확인해야 한다. 본 분석은 뒷부분의 다공선성을 확인을 하기 때문에 앞의 상관분석에서는 넘어간다.



<회귀분석 변수선정을 위한 상관분석>

추정

다음의 단계는 회귀모형의 추정 단계다. 앞서 살펴본 결과에 따라 Y에 영향력이 적은 변수 X3, X4, X5, X7, X8 변수를 제거하고 남은 변수들로 회귀식을 추정한다. 추정의 단계는 변수선택법을 사용해 변수를 선정한 뒤 모형을 추정한다. 변수선택법에는 3가지가 존재한다. 전진선택법, 후진제거법, 단계적 선택법이다. 전진선택법은 변수를 하나씩 더하며 추정하는 모형이고, 후진제거법은 모든 변수를 선택한 상황에서 하나씩 제거하는 방법이다. 단계적 선택법은 변수를 하나씩 더하고 넣고 하는 과정을 통해 최적의 모형을 추정하는 변수선택법이다. 본 분석은 단계적 선택법을 적용해 분석을 실시한다.

```
## Call:
## lm(formula = Y ~ X1 + X5 + X8 + X11 + X12, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24619.9  -4377.9   -297.8   4237.3  23828.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10991.6910   776.6671  14.152  < 2e-16 ***
## X1              1.4690     0.2653   5.538 5.42e-08 ***
## X5             -2.6582     0.6896  -3.855 0.000134 ***
## X8            341.2951    83.5621   4.084 5.30e-05 ***
## X11          -330.4275   124.9721  -2.644 0.008501 **
## X12          1830.0695   174.0208  10.516  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7152 on 418 degrees of freedom
## Multiple R-squared:  0.4315, Adjusted R-squared:  0.4247
## F-statistic: 63.45 on 5 and 418 DF,  p-value: < 2.2e-16
```

검정

실시한 결과에 따라 도출된 분산분석표의 p-value가 유의수준 a(0.05)를 넘는 변수는 제거하고 난 뒤, 모형을 다시 추정한다.

검정결과 단계적 선택법 결과 독립변수'X1, X5, X8, X11, X12'가 남는다. 해당하는 변수들의 회귀분석결과는 다음과 같다. 최솟값 = -24619.9, 1분위수 = -4377.9, 중앙값 = -297.8, 3분위수 = 4237.3, 최댓값 = 23828.1을 갖는 범위를 갖는다. X1 계수는 1.469로 검정통계량은5.538이다. 이때의 p-value = 5.42e-08로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서,

X1은 위 회귀모형에 적합하다고 할 수 있다.

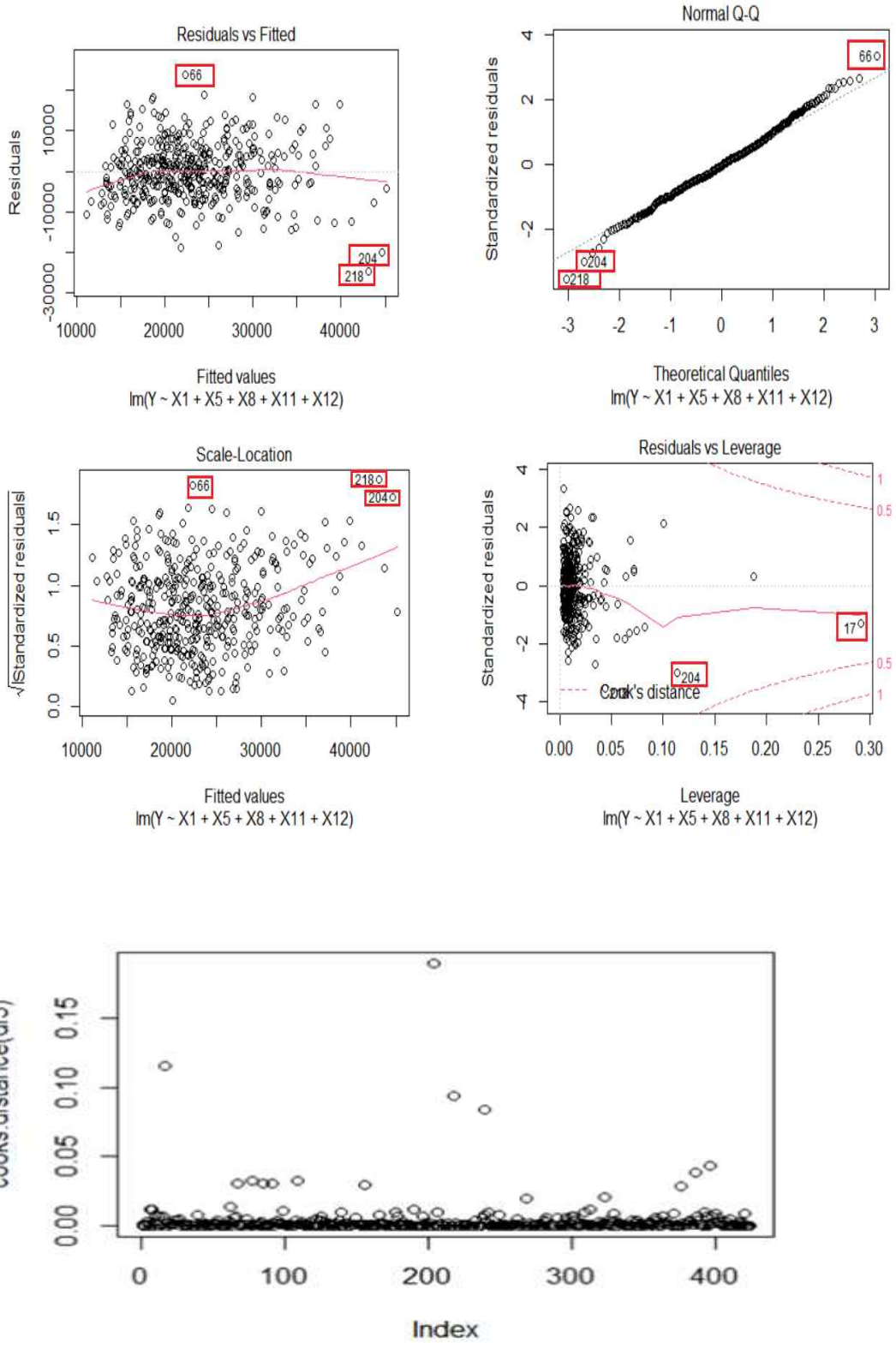
X5 계수는 -2.6582로 검정통계량은 -3.855이다. 이때의 p-value = 0.000134로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X5은 위 회귀모형에 적합하다고 할 수 있다.

X8 계수는 341.2951로 검정통계량은 4.084이다. 이때의 p-value = 5.30e-05로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X8은 위 회귀모형에 적합하다고 할 수 있다.

X11 계수는 -330.4275로 검정통계량은 -2.644이다. 이때의 p-value = 0.008501로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X11은 위 회귀모형에 적합하다고 할 수 있다.

X12 계수는 1830.0695로 검정통계량은 10.516이다. 이때의 p-value = 2e-16로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X12은 위 회귀모형에 적합하다고 할 수 있다.
위 회귀식의 결정계수( )는 0.43155이고, 수정된 결정계수(djR^2)는 0.4247이다.

다음은 회귀모형의 가정사항을 충족시키는지 확인하는 잔차산점도와Q-Q플롯, 표준화 잔차, 레버리지 잔차, 쿡의 거리측도를 살펴보며 잔차의 정규성, 독립성, 등분산성을 위배하는지 살펴본다.


Residuals vs Fitted
Residuals
Fitted values
lm(Y ~ X1 + X5 + X8 + X11 + X12)
Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Y ~ X1 + X5 + X8 + X11 + X12)
Scale-Location
√|Standardized residuals|
Fitted values
lm(Y ~ X1 + X5 + X8 + X11 + X12)
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(Y ~ X1 + X5 + X8 + X11 + X12)
cooks.distance(df3)
Index

<가정사항 그래프>

가정사항

1단계 : 플롯은 모형의 잔차산점도로 잔차가 추정값의 변화에 관계없이 0을 중심으로 고르게 퍼짐을 확인한다. 잔차의 분산이 일정하며 추정값이 증가함에 잔차의 산포는 일정해야 한다. 이것을 등분산성이라 한다. 등분산성을 위배한다면 회귀모형이 적합하지 않다.
따라서, 위 플롯은 66번, 204번, 218번의 잔차가 크다는 것은 확인이 가능하나, 전체적인 모형은 독립성과 등분산성을 띄는 것을 알 수 있다.

2단계 : 두 번째는 Q-Q플롯으로 잔차의 정상성을 확인 할 수 있는 정규확률도로 오차항이 정규분포를 따르는지를 검정하기 위해 사용되는 방법이다. 수직축은 잔차를 나타내고 수평축은 잔차에 대환 누적확률을 표준정규확률변수의 값으로 환산하여 나타낸다. 이 때, 각 점들이 직선에 가까울수록 오차가 정규분포를 따르는 것으로 판단한다.
따라서, 66번, 204번, 218번의 잔차가 선에서 벗어나지만 전체적으로 선형을 띄는 것을 확인할 수 있으므로 오차는 정규분포를 따르는 것으로 확인된다.

3단계 : 세 번째 플롯은 표준화 잔차를 나타내는 그래프로 잔차를 해당 표준편차의 추정치로 나눈 값이다. 일반적으로 2보다 크고 -2보다 작은 표준화 잔차를 큰 값으로 간주한다.
따라서, 앞에서 관측된 66번, 204번, 218번의 잔차가 2를 벗어나는 잔차로 측정이 되므로 제거하는 잔차로 선정한다.

4단계 : 네 번째 플롯은 레버리지 잔차를 표현하는 그래프다. 레버리지 잔차는 잔차가 예측 모형에 벗어나 영향을 주는 잔차를 찾아내는 지표로
현재 출력된 그래프로 판단하기 어려우므로 쿡의 거리측도를 출력해 판단하는 것이 적합하다.

```
################이상치 확인 제거 후 모델 확인################

df4 <- df[-c(66,204,218),]  # 이상치 행 제거.
df5 <- lm(Y~X1+X5+X8+X11+X12,data=df4) # 이상치 행 제거 후 회귀모형
summary(df5)  # 회귀모형 확인

##
## Call:
## lm(formula = Y ~ X1 + X5 + X8 + X11 + X12, data = df4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19492.2  -4540.9   -249.5   4209.5  18821.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10134.6407   765.4394  13.240  < 2e-16 ***
## X1              1.4276     0.2571   5.553 5.02e-08 ***
## X5             -2.6934     0.6686  -4.028 6.68e-05 ***
## X8            442.5961    83.0233   5.331 1.61e-07 ***
## X11          -349.5491   120.4474  -2.902   0.0039 **
## X12          1812.6799   167.5715  10.817  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6885 on 415 degrees of freedom
## Multiple R-squared:  0.4691, Adjusted R-squared:  0.4627
## F-statistic: 73.35 on 5 and 415 DF,  p-value: < 2.2e-16
```

회귀식 도출

검정결과 독립변수'X1, X5, X8, X11, X12'의 회귀분석 결과는 다음과 같다 최솟값 = -19492.2, 1분위수 -4540.9, 중앙값 = -249.5, 3분위수 = 4209.5, 최댓값 = 18821.3 을 갖는 범위를 갖는다.

X1 계수는 1.4276로 검정통계량은5.533이다. 이때의 p-value 2e-16로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X1은 위 회귀모형에 적합하다고 할 수 있다.

X5 계수는 -2.6934로 검정통계량은 -4.028이다. 이때의 p-value = 6.68e-05 로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X5은 위 회귀모형에 적합하다고 할 수 있다.

X8 계수는 341.2951로 검정통계량은 5.331이다. 이때의 p-value = 1.61e-07로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X8은 위 회귀모형에 적합하다고 할 수 있다.

X11 계수는 -349.5491로 검정통계량은 -2.902이다. 이때의 p-value = 0.0039로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X11은 위 회귀모형에 적합하다고 할 수 있다.

X12 계수는 1812.6799로 검정통계량은 10.817이다. 이때의 p-value = 2e-16로 유의수준 0.05를 기각하며, 매우 유의한 것으로 나온다. 따라서, X12은 위 회귀모형에 적합하다고 할 수 있다.

결론

위 분석 결과 추정 된 회귀식은 다음과 같다.

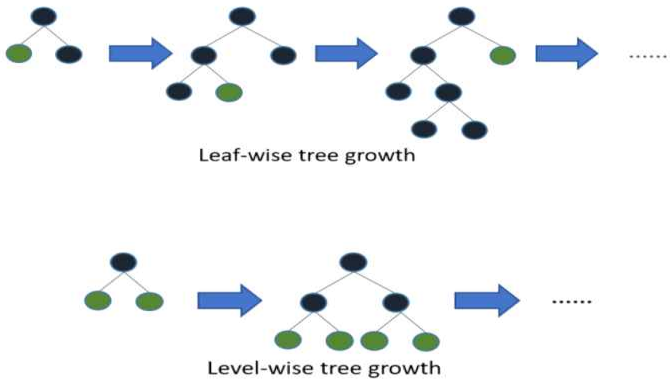
$$1.4276*x\ -2.6934*x_5+341.2951*x_8-349.5491*x_{11}+1812.6799*x_{12}$$

추정된 회귀식의 결정계수(R^2)는 0.4691이고, 수정된 결정계수($adj\,R^2$)는 0.4627이다.

회귀분석은 중심극한정리에 의하여 평균과 분산이 존재하는 모집단 N이 충분히 클 때, 표본평균 x는 평균이 이고, 분산이 $\frac{\sigma^2}{N}$을 따르고, 표본평균 $\bar{x}$를 표준화시키면 평균이 0이고, 분산이 1인 표준정규분포를 따르는데 추출된 표본의 수를 더욱 추출하거나 표본추출방법을 새롭게 하거나 새로운 변수를 찾는다면 더욱 정교한 모형의 추정이 가능할 것으로 보인다.

Light GBM

LightGBM 모델은 기존 Gradien Boosting Machine(GBM: 모델에서 정답지와 오답지간의 차이를 경사하강법을 사용하여 훈련에 다시 투입하여 gradient를 적극 이용해서 모델을 개선하는 알고리즘) 모델에서 일반적인 균형트리분할(Level Wise) 방식과 달리 리프중심 트리분할(Leaf Wise) 방식을 사용한다. 트리가 깊어지고 비대칭적으로 생성한다. 이로써 예측 오류 손실을 최소화 한다.



<Light GBM 모형>

상관분석을 거쳐 산출된 최종 데이터 셋에서 'Degree'를 종속변수로 설정하고 사전에 선정한 12개의 컬럼을 독립변수로 설정한다.
StandardScaler를 사용하여 독립변수(피쳐)를 수치데이터로 비교하기 쉽게 정규화 한다.

SMOTE으로 데이터 개수가 적은 표본을 임의의 값을 추가하여 오버샘플링을 진행하고 데이터 전처리 과정을 마무리한다.
모델을 학습시키기 위해서 Train set과 test셋을 7:3의 비율로 나눠서 진행합니다.

Model Learning 전, LightGBM의 파라미터인 반복 수행하는 트리의 개수, 트리가 가질 수 있는 최대 리프 개수, 트리의 최대 깊이를 최적의 하이퍼 파라미터를 구하기 위해 GridSearchCV를 진행한다.

최적의 하이퍼 파라미터로 best model을 생성하고 Train set으로 모델을 훈련시키고 Test set으로 모델의 성능을 검증한다.

랜덤포레스트

서울시 내에 있는 각 동(읍면동)의 오토바이 수요 정도(1~4단계)를 예측하는 Model을 랜덤포레스트 기법을 사용하여 modeling한다.

오토바이 수요의 간접적 변수로 정한 주민등록인구의 분포를 boxplot으로 확인한다. boxplot상의 사분위수 기준에 따라 오토바이 수요 정도를 나눈다. 주민등록인구를 기준으로 16,937명 미만, 16,937명 이상이고 22,329명 미만, 22,329명 이상이고 28,813.5명 미만, 28,813.5명 이상으로 4단계로 걸쳐 나눈다. 이 기준에 따라 파생변수 "Degree"를 생성하여 사전에 선정한 12개의 독립변수로 구성된 데이터 셋에 추가한다.

Train set과 Test set을 7:3 비율로 나누며, 이 과정을 수행할 때마다 결과가 다르게 나눠지지 않도록 고정 seed값을 지정한다. 이전에 추가한 파생변수 "Degree"를 factor형으로 변경하여 랜덤포레스트 model에 바로 사용할 수 있도록 전처리 과정을 마무리한다.

랜덤포레스트 Model을 modeling할 때 필요한 최적의 하이퍼 파라미터를 구하기 위해 그리드 검색을 수행한다. 찾고자 하는 최적의 하이퍼 파라미터로는 트리 개수, 의사결정 나무 분기에 사용되는 변수 개수, leaf node(terminal node)에 담기는 표본 크기가 있다. tune함수로 다양한 하이퍼 파라미터 경우의 수에 대해 그리드 검색을 수행한다.

최적의 하이퍼 파라미터로 best_model을 생성하고 Train set과 Test set을 사용하여 model을 data에 fit하게하고 성능을 검증한다. 성능을 평가하는 척도로 정확도(accuracy)와 F1-score를 채택한다. 정확도는 $\frac{TP+TN}{tal}$ 이다.

F1-score는 Precision과 Recall의 조화평균이다. Precision(정밀도)는 $\frac{TP}{TP+FP}$ 이고 Recall(재현율)은 $\frac{TP}{P+FN)}$ 이다. 따라서 F1-score는 조화평균 값으로 $2*\frac{\mathrm{P}recision*recall}{\mathrm{P}recision+recall}$ 이다. 정확도 역시 model의 중요한 평가 척도이지만 우리가 예측하고자 하는 목표를 잘 수행하는지 평가하는 F1-score도 매우 중요한 평가 척도이다.

4. 분석결과

가. 진행한 분석 내용 순서대로

1. 다중 회귀분석 결과

```
#################이상치 확인 제거 후 모델 확인#################

df4 <- df[-c(66,204,218),]  # 이상치 행 제거.
df5 <- lm(Y~X1+X5+X8+X11+X12,data=df4) # 이상치 행 제거 후 회귀모형
summary(df5)  # 회귀모형 확인

##
## Call:
## lm(formula = Y ~ X1 + X5 + X8 + X11 + X12, data = df4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19492.2  -4540.9   -249.5   4209.5  18821.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10134.6407   765.4394  13.240  < 2e-16 ***
## X1              1.4276     0.2571   5.553 5.02e-08 ***
## X5             -2.6934     0.6686  -4.028 6.68e-05 ***
## X8            442.5961    83.0233   5.331 1.61e-07 ***
## X11          -349.5491   120.4474  -2.902   0.0039 **
## X12          1812.6799   167.5715  10.817  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6885 on 415 degrees of freedom
## Multiple R-squared:  0.4691, Adjusted R-squared:  0.4627
## F-statistic: 73.35 on 5 and 415 DF,  p-value: < 2.2e-16
```

<회귀분석 최종 결과>

결론

위 분석결과 추정된 회귀식은 다음과 같다.

$$1.4276*x\ -2.6934*x_2+341.2951*x_8-349.5491*x_{11}+1812.6799*x_{12}$$

추정된 회귀식의 결정계수(R^2)는 0.4691이고, 수정된 결정계수($adj\,R^2$)는 0.4627이다.

회귀분석은 중심극한정리에 의하여 평균과 분산이 존재하는 모집단 N이 충분히 클 때, 표본평균 x는 평균이 μ이고, 분산이 $\frac{\sigma^2}{N}$을 따르고, 표본평균 $\bar{x}$를 표준화시키면 평균이 0이고, 분산이 1인 표준정규분포를 따르는데 추출된 표본의 수를 더욱 추출하거나 표본추출방법을 새롭게 하거나 새로운 변수를 찾는다면 더욱 정교한 모형의 추정이 가능할 것으로 보인다.

2. Light GBM

<X값 정규화 전>

df_data_x - DataFrame

Index	X2	X3	X4	X5	X6	X7	X8	X9	x10
0	806	194	1096	49	75	16	25	0	4
1	624	690	3032	120	295	119	360	8	9
2	1939	2034	11667	705	713	205	637	10	19
3	6619	5269	40662	1309	2328	779	3858	18	100
4	4246	5349	58471	1157	2344	886	4344	15	131
5	389	1098	6938	162	588	155	959	10	18
6	1605	4141	62562	1062	1794	522	2764	16	88
7	3958	3639	28207	710	1618	686	3245	25	101
8	1726	1527	17284	339	669	225	1025	8	40
9	1365	1825	21955	475	747	255	1386	4	39
10	1289	3921	61996	1323	1284	648	5998	14	88
11	3908	3610	40771	730	1585	488	2301	17	73

Format Resize Background color Column min/max Save and Close Close

<X값 정규화 후>

x_data - NumPy object array

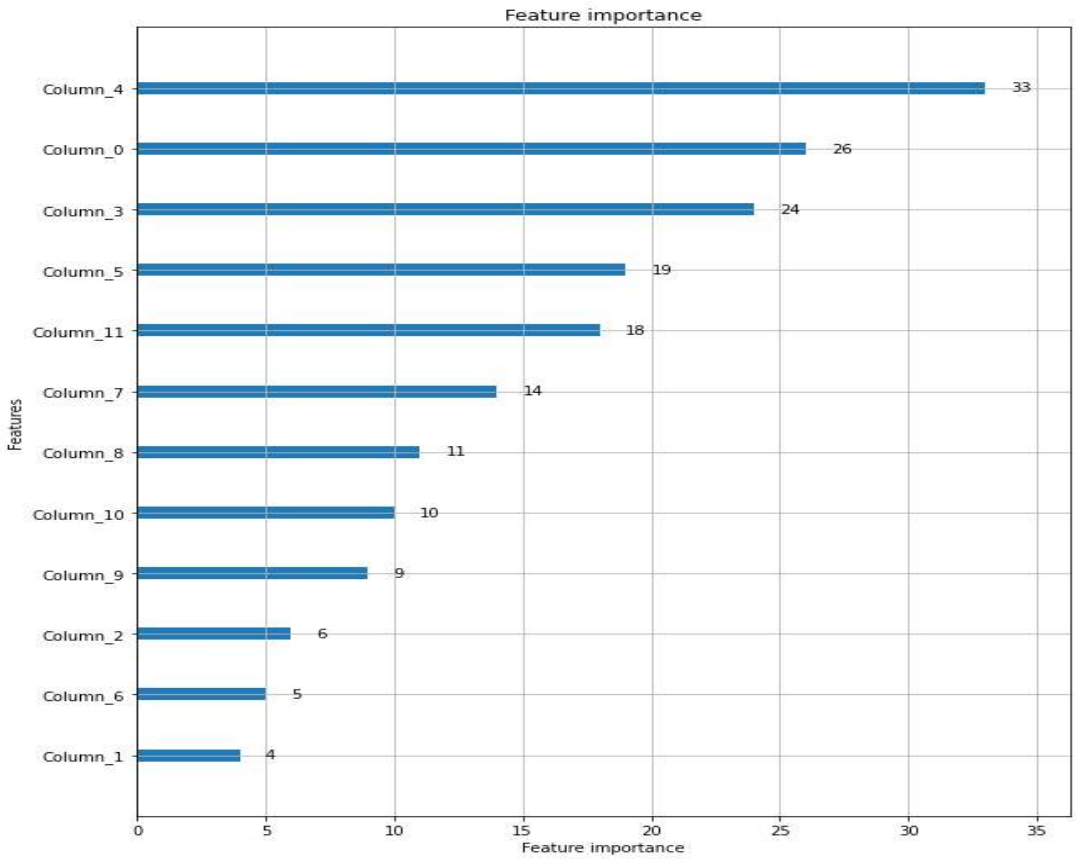
	0	1	2	3	4	5	6
0	-1.14248	-0.807417	-0.58393	-0.53464	-0.76888	-1.00644	-0.748171
1	-1.24179	-0.548262	-0.481077	-0.455425	-0.42132	-0.65144	-0.528828
2	-0.524192	0.153963	-0.0223304	0.197261	0.239044	-0.355029	-0.34746
3	2.02971	1.84422	1.51807	0.871145	2.79045	1.62335	1.76151
4	0.73475	1.88601	2.4642	0.701558	2.81573	1.99214	2.07973
5	-1.37004	-0.335087	-0.273565	-0.408566	0.0415671	-0.527361	-0.136628
6	-0.706458	1.25485	2.68154	0.595567	1.94683	0.737558	1.04521
7	0.577586	0.992558	0.856381	0.202839	1.66878	1.30281	1.36015
8	-0.640428	-0.110939	0.276081	-0.211086	0.169532	-0.286096	-0.0934141
9	-0.837427	0.0447628	0.524234	-0.0593506	0.292758	-0.182697	0.142953

Format Resize Background color Save and Close Close

1) 하이퍼 파라미터 도출

반복트리의 개수는 100, 200, 1000개 중에서, 트리가 가질 수 있는 최대 리프 개수는 10, 20, 50 중에서, 트리의 최대 깊이는 10, 20, 50 중에서 GridSearchCV를 사용하여 최적의 조합을 찾는다.
최적의 하이퍼 파라미터는 반복트리의 개수는 100개, 최대 리프개수는 10개, 트리의 최대 깊이는 10으로 설정한다.

2) 변수 중요도



<변수 중요도>

12개의 독립변수의 중요도를 출력한다. 출력한 결과 Column_4(X6)가 제일 높게 나왔으며 Column_1(X3)이 제일 낮다. 따라서 Column_7(X9)보다 낮은 컬럼들은 종속변수를 잘 설명하지 못한다고 판단할 수 있다.

3) 모델검증

```
In [4]: estimator = grid_model.best_estimator_
   ...: y_predict = estimator.predict(x_test)
   ...: print("train score: {:.3f}".format(estimator.score(x_train, y_train)))
   ...: print("test score: {:.3f}".format(estimator.score(x_test, y_test)))
train score: 0.628
test score: 0.508
```

Test set으로 모델을 검증한 결과 50.8%로 나타났다. 값이 낮은 이유는 데이터를 전처리 한 결과 데이터 양이 감소하여서 낮게 나온 것으로 보인다. LightGBM 모델의 권장사항으로 10,000건 이상의 데이터 셋 사용 권장을 LightGBM 공식 문서에 기술하고 있다.

```
In [5]: from sklearn.metrics import classification_report
   ...: rep = classification_report(y_test, y_predict)
   ...: print(rep)
              precision    recall  f1-score   support

           1       0.68      0.81      0.74        32
           2       0.46      0.34      0.39        32
           3       0.36      0.31      0.33        32
           4       0.47      0.56      0.51        32

    accuracy                           0.51       128
   macro avg       0.49      0.51      0.50       128
weighted avg       0.49      0.51      0.50       128
```

모델의 정확도는 낮게 나왔지만 F1-score를 확인하여 모델의 적합성을 확인하기 위해 출력해보니 1등급 F1-score 외 모두 낮게 나와서 LightGBM 모델은 의미가 없는 것으로 판단된다.

3. 랜덤포레스트

1) 최적 하이퍼 파라미터 도출

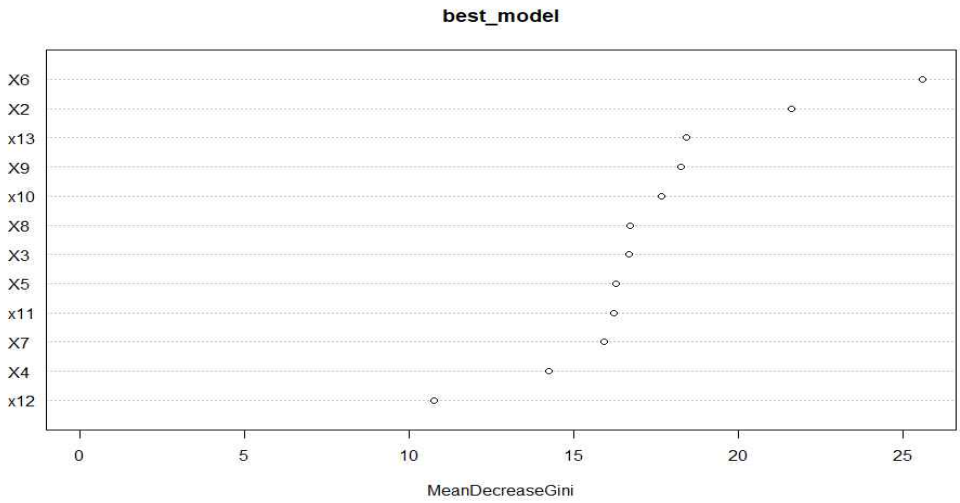
모델의 안정성을 보장하기 위해 e1071 라이브러리의 tune함수로 그리드 검색을 수행하여 Train set을 바탕으로 5중 교차 검증을 실시한다. 트리 개수는 100, 200, 300, 400개 중에서, 의사결정 나무 분기에 사용되는 변수 개수는 3, 4, 5개 중에서, leaf node(terminal node)에 담기는 표본 크기는 1, 2개 중에서 최적의 조합을 찾는다.

```
summary(rf_grid)

##
## Parameter tuning of 'randomForest':
##
## - sampling method: 5-fold cross validation
##
## - best parameters:
##  mtry ntree nodesize
##     3   100        2
##
## - best performance: 0.5471751
##
## - Detailed performance results:
##    mtry ntree nodesize     error dispersion
## 1     3   100        1 0.5844068 0.03957545
## 2     4   100        1 0.5741243 0.04424201
## 3     5   100        1 0.5976836 0.05354445
## 4     3   200        1 0.5911299 0.02047545
## 5     4   200        1 0.5809605 0.03421206
## 6     5   200        1 0.5875706 0.05761604
## 7     3   300        1 0.5709605 0.02518279
## 8     4   300        1 0.5741808 0.03008680
## 9     5   300        1 0.5674576 0.03453637
## 10    3   400        1 0.5641243 0.02887028
## 11    4   400        1 0.5606215 0.04673908
## 12    5   400        1 0.5707345 0.05980487
## 13    3   100        2 0.5471751 0.02461845
## 14    4   100        2 0.5741808 0.05659864
## 15    5   100        2 0.5910734 0.03229346
## 16    3   200        2 0.5741808 0.03655306
## 17    4   200        2 0.5571751 0.05620603
## 18    5   200        2 0.5674011 0.03724081
## 19    3   300        2 0.5741243 0.05309592
## 20    4   300        2 0.5841808 0.05109794
## 21    5   300        2 0.5606780 0.04128084
## 22    3   400        2 0.5571751 0.04940580
```

<최적 파라미터 도출>

최적 하이퍼 파라미터에 대한 분석 결과는 다음과 같다. 의사결정 나무 분기에 사용되는 변수 개수는 3개, 트리 개수는 100개, leaf node(terminal node)에 담기는 표본 크기는 1개로 결정되었다.



<중요도 변수>

12개의 설명변수의 중요도를 파악한다. 변수 중요도 그래프는 가장 중요한 변수들의 순서를 지니 값 감소의 평균에 따라 내림차순으로 정렬해 나타낸다. 그래프에서 높이 있는 변수 일수록, 모델에 더 많은 기여를 한다. X6(서비스업 등록 수) 가장 중요한 변수이며, X2(1인 가구 수), X13(슈퍼마켓 수)가 이어서 중요한 변수로 나타났다. 이와 반대로 X4(사업체 종사자 수), X12(PC방 수)는 상대적으로 중요하지 않은 변수로 판단된다.

```
##    y_pred_train
##       1  2  3  4
##   1 48 15  7  3
##   2 20 27 21 10
##   3  4 19 28 26
##   4  3 12 25 28
#model 성능의 평가 척도 중, 정확도(accuracy)를 계산 및 출력
train_acc <-
(train_conf_mat[1,1]+train_conf_mat[2,2]+train_conf_mat[3,3]+train_conf_mat[4
,4])/sum(train_conf_mat)
print(paste("Train Confusion Matrix - Grid Search Accuracy:",
round(train_acc,4)))
## [1] "Train Confusion Matrix - Grid Search Accuracy: 0.4426"
```

<첫 번째 Train set에 best model>

이전 단계의 최적 하이퍼 파라미터로 생성한 best model을 Train set에 적용한다. 혼동행렬(confusion matrix)는 다음과 같이 나타난다. 혼동행렬을 바탕으로 (혼동행렬의 [1,1]+[2,2]+[3,3]+[4,4]) / (혼동행렬 행렬 전체 합)을 계산하여 정확도(accuracy)를 구한다. Train set을 입력한 best model의 정확도는 44.26%로 나타났다.

```
#F1-score 계산
F1_Score(train_set$Degree, y_pred_train, positive = NULL)

## [1] 0.6486486
```

<그림1. 첫 번째 F1-score 계산>

반면, F1-score는 0.6486486으로 나타났다. F1-score는 Precision과 Recall의 조화평균으로 계산되고, 이는 precision과 recall이 0에 가까울수록 F1-score도 동일하게 낮은 값을 갖도록 하기 위함이다. 따라서 F1-score는 높을수록 Precision, Recall이 높다는 것을 의미하고, 이는 모델의 성능이 우수하다는 것을 뜻한다. 사용한 data가 imbalanced data인 것을 감안했을 때, model이 결과를 잘 예측했는가를 평가함에 있어서 본 model이 유의한 것으로 판단된다.

```
print(test_conf_mat)

##    y_pred_test
##       1  2  3  4
##   1 27  2  3  1
##   2  7  9  6  6
##   3  5  9 10  5
##   4  1  7  8 22

#model 성능의 평가 척도 중, 정확도(accuracy)를 계산 및 출력
test_acc <-
(test_conf_mat[1,1]+test_conf_mat[2,2]+test_conf_mat[3,3]+test_conf_mat[4,4])
/sum(test_conf_mat)
print(paste("Test Confusion Matrix - Grid Search Accuracy:",
round(test_acc,4)))

## [1] "Test Confusion Matrix - Grid Search Accuracy: 0.5312"
```

<두 번째 Train set에 best model>

최적 하이퍼 파라미터로 생성한 best model을 Test set에 적용한다. 혼동행렬(confusion matrix)는 다음과 같이 나타난다. 같은 방법으로 (혼동행렬의 [1,1]+[2,2]+[3,3]+[4,4]) / (혼동행렬 행렬 전체 합)을 계산하여 정확도(accuracy)를 구한다. Test set을 입력한 best model의 정확도는 53.12%로 나타났다. 이는 Train set을 입력한 경우의 정확도보다 8.86%p 증가한 것을 알 수 있다.

```
#F1-score 계산
F1_Score(test_set$Degree,  y_pred_test, positive = NULL)
## [1] 0.739726
```

<두 번째 model F1- score 계산>

F1-score는 0.739726으로 Train set의 것보다 약 0.091 정도 높은 것으로 나타났다. 전반적으로 정확도와 F1-score가 Train set의 것보다 더 높은 수치를 기록해 의미있는 best model을 modeling했다고 판단한다.

최종 모델 선정

최종모델 선정은 랜덤포레스트로 한다. 다중회귀분석, LightGBM의 모델의 단점들은 다음과 같다. 다중회귀 분석은 모델과 멀리 떨어져 있는 잔차들을 삭제함으로써 정확도를 올린다. 이에 각 표본의 특성을 무시할 가능성이 높을 뿐더러 중요한 변수를 선정하는 과정에서 연구자의 견해가 들어간다. LightGBM 모델은 10,000개 이상의 표본이 존재했을 때, 분류와 정확도가 정확하기에 현재 서울시 행정동을 대상으로는 적합하지 않다. 이러한 점들을 보완할 수 있는 랜덤포레스트 모델을 채택한다.

5. 활용 방안

가. KPI(Key Performance Indicator)

• 배터리 교체소 인프라 구축으로 전기 오토바이 전년 대비 대략 2배 증가량이 있을 것으로 예상한다.

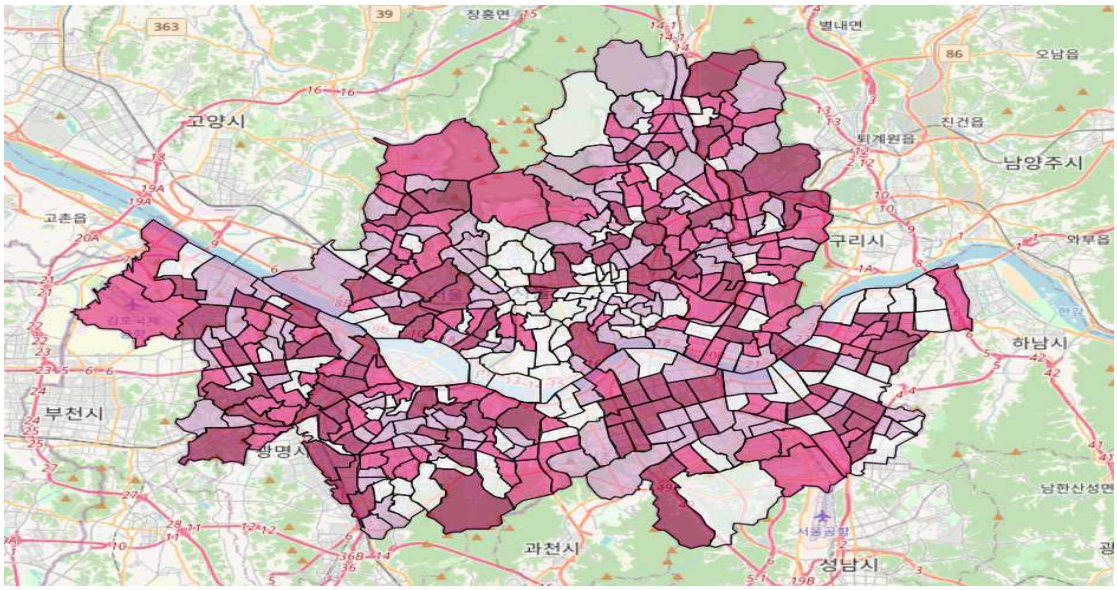
그 이유는 2018년도까지 지지부진하던 전기 오토바이 구매량이 2019년도에 급증했던 이유는 소비자들에게 가격적인 메리트가 와 닿았기 때문이다.
따라서 전기 오토바이 배터리 교체소의 인프라가 확충된다면 잠재고객들과 기존 오토바이 사용 고객들의 needs와 wants를 만족시키며 2019년도의 증가량과 비슷할 것으로 예상하기 때문이다.

• 단기 : 충전 인프라 입지선정 모형 활용
- 입지선정 요건에 대한 과학적이고 객관적인 지표 제공을 통한 지방 자치 예산의 효율적 사용
- 전기 오토바이 배터리 교체소 이용에 따른 사용자의 불편을 최소화하여 전기 오토바이 사용 촉진을 위한 홍보 매개체로 활용

• 장기 : 충전 인프라 입지선정 모형 고도화 및 전기 오토바이 신청 수요 예측
- 충전 인프라 환경의 향후 변동성을 고려한 확대 적용 예측모델 및 입지선정모델 고도화 추진
- 전기 오토바이 이미지 개선과 제약요건 해소를 통해 전기 오토바이 수요증가 예측

나. 문제점 개선 방안

앞서 언급한 전기오토바이의 확산을 위해 서울시 內 배터리 교체소의 충전이 시급한 지역을 선정하는 청사진을 제시한다.



<예측모델 지도>

<기존 설치된 교환소 위치>

기존에 설치된 배터리 교환소(빨간색 지점)와 본 프로젝트의 모델이 예측한 오토바이 수요가 가장 높은 지역(4단계)를 비교한다.

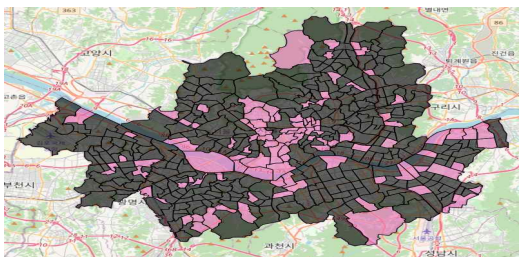
<기존 설치 교환소 위치와 예측모델 비교>

기존 설치된 교환소 위치	모델이 예측한 오토바이 수요도	결과 비교
역삼1동	4단계	매우적합
용신동	4단계	매우적합
논현2동	3단계	적합
면목본동	3단계	적합
송파1동	3단계	적합
방배2동	3단계	적합
삼전동	3단계	적합
중곡3동	3단계	적합
대치4동	2단계	조금적합
지앙2동	2단계	조금적합
도림동	2단계	조금적합
신림동	2단계	조금적합
역삼2동	2단계	조금적합

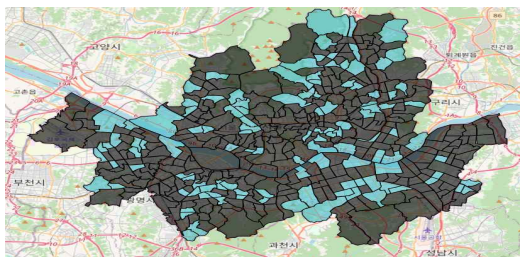
Python의 folium, matplotlib.pyplot 라이브러리를 이용하여 모델이 예측한 오토바이 수요도를 서울시 지도에 시각화 표현함.

해당 모델을 이용하면 전기 오토바이 배터리 교체소를 시급한 지역에 따라 해당 구역을 알 수 있다.

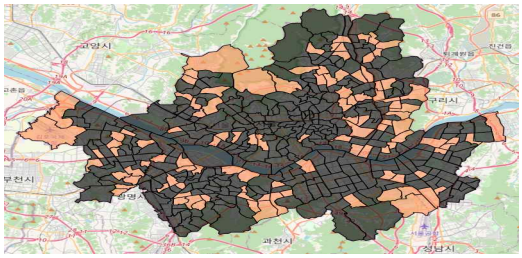
'서울시'의 스마트 모빌리티 보급 사업에서 해당 예측 모델을 사용한다면 보다 효율적인 운용이 가능할 것이다.



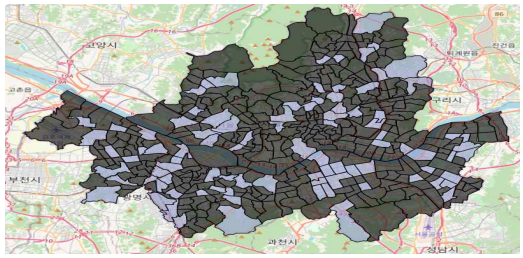
<1단계>



<2단계>



<3단계>



<4단계>

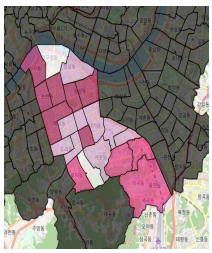
<서울시 단계별 수요지 예측>

<각 구 별 전기 오토바이 예상수요 [단위 : 동(개)]>

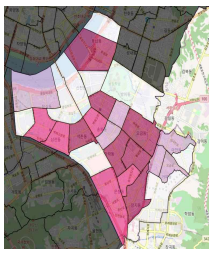
구	4단계 수요지	3단계 수요지	2단계 수요지	1단계 수요지
강남구	7	5	9	2
송파구	7	4	6	9
구로구	6	5	2	2
성북구	6	2	8	4
강동구	5	7	1	6
강서구	5	6	6	3
노원구	5	6	4	3
동대문구	5	5	4	0
마포구	5	4	4	5
서초구	5	4	4	3
관악구	4	8	5	4
동작구	4	5	6	2
양천구	4	3	5	4
광진구	3	7	4	1
금천구	3	6	7	2
서대문구	3	6	4	2
영등포구	3	2	5	4
은평구	3	1	5	2
종로구	2	1	0	14
강북구	1	8	5	2
도봉구	1	5	8	5
성동구	1	5	6	2
용산구	1	3	3	10
중구	1	2	2	11
중랑구	1	1	4	3

4단계 크기순으로 정렬 후 동점 수는 3단계, 2단계, 1단계 크기 순으로 정렬함.

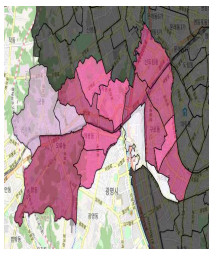
<각 구별 설치 적합 동 표시>



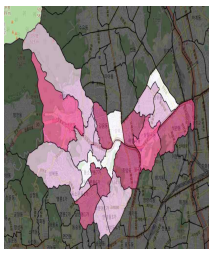
<강남구>



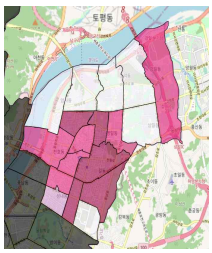
<송파구>



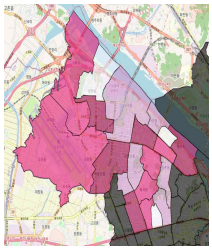
<구로구>



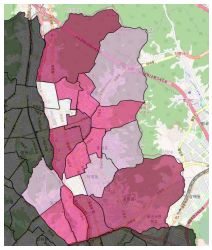
<성북구>



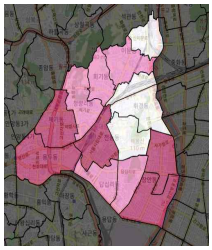
<강동구>



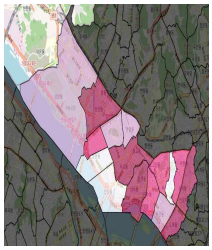
<강서구>



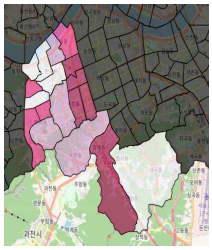
<노원구>



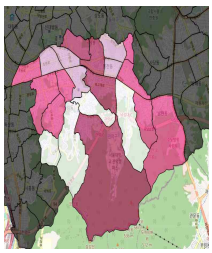
<동대문구>



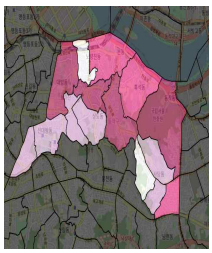
<마포구>



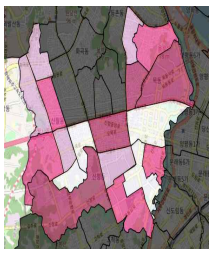
<서초구>



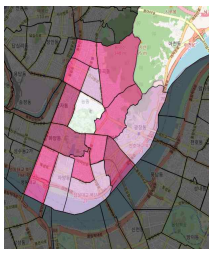
<관악구>



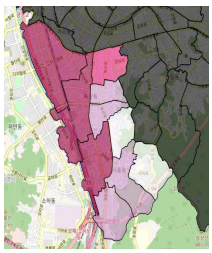
<동작구>



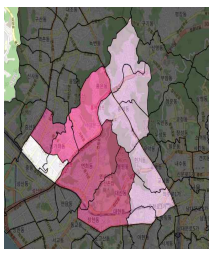
<양천구>



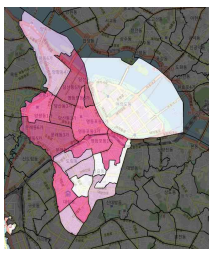
<광진구>



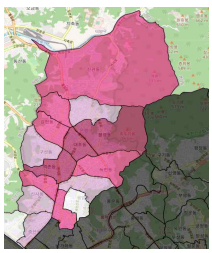
<금천구>



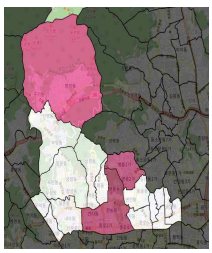
<서대문구>



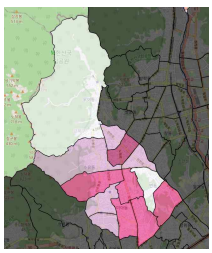
<영등포구>



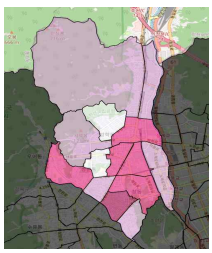
<은평구>



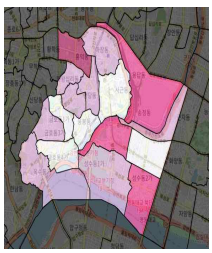
<종로구>



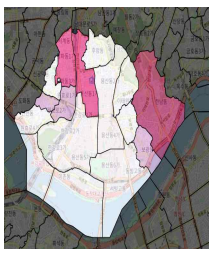
<강북구>



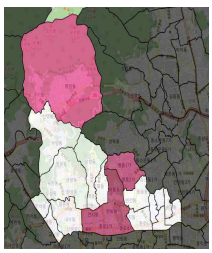
<도봉구>



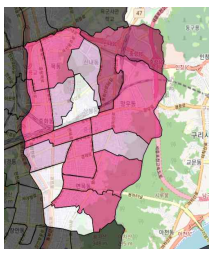
<성동구>



<용산구>



<중구>



<중랑구>

다. 업무 활용 방안

1) 배터리 교체소 설치 시범사업에 기여

분석 결과를 바탕으로 내년 계획된 환경부의 전기 오토바이 배터리 교체소 설치 시범사업에 활용하여 사업의 완성도를 높일 수 있다.

2) 그린 뉴딜 사업에 기여

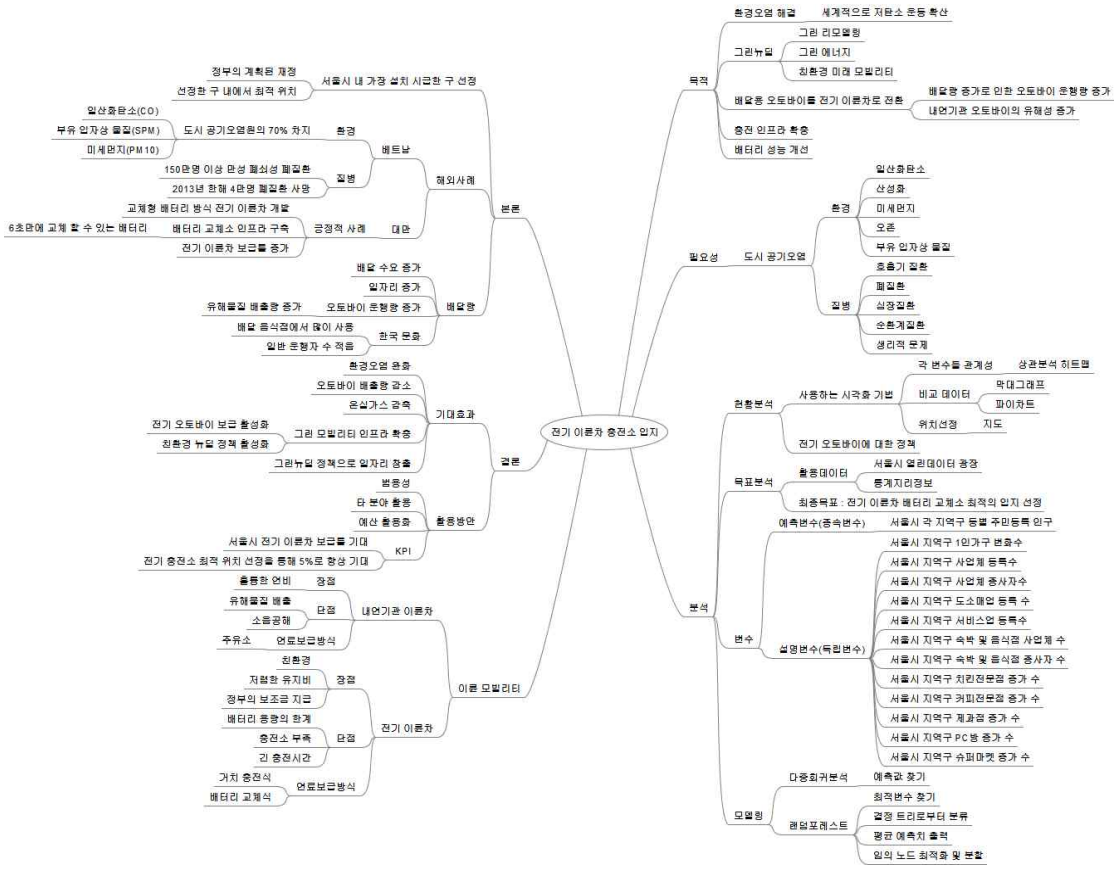
전기 오토바이 배터리 교체소 인프라를 확충하여 친환경 미래 모빌리티 보급 활성화에 기여 하고, 스마트 모빌리티와 배터리 산업 등 경제 활성화를 통한 일자리 창출을 도모할 수 있다.

3) 기존 입지 선정 기준과 본 프로젝트 모델 비교

기존에 설치된 배터리 교체소의 입지선정 기준과 본 프로젝트의 모델이 선정한 입지선정
기준을 비교/활용하여 최적의 입지를 선정한다.

6. [부록]

가. 주제설계를 위한 마인드맵



<주제설계를 위한 마인드맵>

나. 분석 상세코드

[R 코드]

1. 다중회귀분석 코드

라이브러리 (1)

```
library(corrplot)
library(MASS)
library(car)

df <-  read.csv('data.csv')
str(df)
```

상관분석 (2)

```
cor_df <-  cor(df)
cor_df
plot(df)
```

회귀모형 추정 (3)

```
df2 <-
lm(Y~X1+X5+X8+X10+X11+X12,data=df)
summary(df2)
stepAIC(df2)

df3 <-  lm(Y~X1+X5+X8+X11+X12, data=df)
summary(df3)
```

이상값, 가정사항 그래프 및 쿡의 거리 측도 (4)

```
vif(df3)
plot(df3)
```

이상치 제거 후 모델 (5)

```
df4<-  df[-c(66,204,218),]
df5<-lm(Y~X1+X5+X8+X11+X12,data=df4)
summary(df5)

plot(df3)
```

2. 랜덤포레스트 코드

파일 불러오기 (1)

```
data <- read.csv("dataset2.csv",    header=T)
```

파생변수 생성 (2)

```
data$Degree <- ifelse(data$Y < 16937, 1,
                      ifelse(data$Y<22329,

ifelse(data$Y< 28813.5, 3, 4)))
```

시드 고정 (3)

```
set.seed(123)

splt <- sample(1:numrow, size =
as.integer(0.7*numrow))

train_set <- data[splt,]
test_set <- data[-splt,]

train_set$Degree <-
as.factor(train_set$Degree)
test_set$Degree <-
as.factor(test_set$Degree)

train_set <- train_set[,-1]
test_set <- test_set[,-1]
```

랜덤포레스트 기법 사용하여 생성 (4)

```
library(randomForest)
library(e1071)
library(MLmetrics)

rf_grid <- tune(randomForest,
train.y=train_set$Degree,
train.x=subset(train_set, select = -Degree),
                  data = train_set, ranges =
list(mtry = c(3,4,5),

      ntree = c(100,200,300,400),

      nodesize = c(1,2)),
                   tunecontrol =
tune.control(cross = 5))

summary(rf_grid)
best_model <- rf_grid$best.model
summary(best_model)
```

변수 중요도 그래프 (5)

```
vari <- varImpPlot(best_model)
print(paste("Variable Importance -    Table"))
print(vari)
```

Train Set (6)

```
y_pred_train <- predict(best_model,
ata=train_set)

train_conf_mat <- table(train_set$Degree,
y_pred_train)

print(paste("Train Confusion Matrix - Grid
Search:"))
print(train_conf_mat)

train_acc <-
(train_conf_mat[1,1]+train_conf_mat[2,2]+trai
n_conf_mat[3,3]+train_conf_mat[4,4])/sum(tr
ain_conf_mat)
print(paste("Train Confusion Matrix - Grid
Search Accuracy:", round(train_acc,4)))

F1_Score(train_set$Degree, y_pred_train,
positive    = NULL)
```

Test Set (7)

```
y_pred_test <- predict(best_model, newdata
=    test_set)

test_conf_mat <- table(test_set$Degree,
y_pred_test)

print(paste("Test Confuision Matrix- Grid
Search:"))
print(test_conf_mat)

test_acc <-
(test_conf_mat[1,1]+test_conf_mat[2,2]+test_c
onf_mat[3,3]+test_conf_mat[4,4])/sum(test_co
nf_mat)
print(paste("Test Confusion Matrix - Grid
Search    Accuracy:", round(test_acc,4)))

F1_Score(test_set$Degree,   y_pred_test,
positive = NULL)
```

[Python 코드]

1. 상관분석 코드

사용한 라이브러리 (1)

```
import pandas    as pd
import numpy    as np
import    matplotlib.pyplot as plt
import    seaborn as sns
```

데이터 불러오기 (2)

```
df =    pd.read_excel('test.xlsx',index_col=0)
```

데이터 확인하기 (3)

```
df.head(5)
```

데이터 시각화 & 이미지파일로 저장 (4)

```
%config
lnlineBackend.figure_format='retina'
plt.figure(figsize=(16,9))

sns.heatmap(
        data=df,
        annot = True,
        fmt = '.02f',
        linewidth=5,
        center = 0
).get_figure().savefig('y변수선정 시각화.png',
dpi=450)
```

2. Light GBM코드

사용한 라이브러리 (1)

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from lightgbm import LGBMClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from lightgbm import plot_importance
```

데이터 불러오기 (2)

```
df_data = pd.read_csv('dataframe.csv', index_col=None)
df_data = df_data.iloc[: , 1:]
```

독립변수, 종속변수 분리 (3)

```
df_data_x = df_data.drop(['Degree', 'X1', 'Y'], axis=1) # Degree = 종속변수(인구수)
df_data_y = df_data['Degree']
```

데이터 전처리- 정규화 (4)

```
from sklearn.preprocessing
import StandardScaler
scaler = StandardScaler()
x_data = scaler.fit_transform(df_data_x)
y_data = df_data_y.to_numpy()
```

데이터 전처리 - 데이터 분할 (5)

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_data,
y_data,
test_size = 0.3,
random_state = 777,
stratify = y_data)
```

데이터 전처리 - 오버 샘플링 (6)

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 0)
x_train, y_train = sm.fit_resample(x_train, y_train)
```

모델 설정 (7)

```
from lightgbm import LGBMClassifier
model = LGBMClassifier(random_state = 0, n_jobs=-1)
params = { "n_estimators":[100, 200, 1000],
"objective": ['multiclass'],
"metric": ['multi_logloss'],
"num_leaves": [10, 20, 50],
"max_depth": [10, 20, 50],
"learning_rate":[0.01, 0.05, 0.1]}
from sklearn.model_selection import GridSearchCV
grid_model = GridSearchCV(model,
param_grid = params,
cv = 5,
n_jobs = -1)
```

모델 학습 (8)

```
grid_model.fit(x_train,
y_train,
early_stopping_rounds = 50,
eval_set = [(x_test, y_test)],
eval_metric='error')
```

모델 검증 (9)

```
print(f`BestParam:{grid_model.best_params}`)
```

모델 예측 (10)

```
estimator = grid_model.best_estimator_
y_predict = estimator.predict(x_test)
print("train score: {:.3f}".format(estimator.score(x_train, y_train)))
print("test score: {:.3f}".format(estimator.score(x_test, y_test)))
```

F1 Score 확인 (11)

```
from sklearn.metrics
import classification_report
rep = classification_report(y_test,y_predict)
print(rep)
```

중요변수 시각화 (12)

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize = (10, 12))
from lightgbm import plot_importance
plot_importance(estimator, ax = ax)
```

3. 공간분석(지도) 코드

사용한 라이브러리 (1)

```
import pandas as pd
import    seaborn as sns
from sklearn    import preprocessing
import folium
import    matplotlib.pyplot as plt
import json
import    warnings
warnings.filterwarnings(action='ignore')

plt.rcParams['font.family']='NanumGothic'
```

전처리 데이터 불러오기 (2)

```
df=pd.read_csv("최종서울전처리.csv",
encoding='euc-kr', header=1, index_col=0)

df.head()

colList=["N1인가구변화","도소매업","서비스
업","사업체수","인구변화","주민등록인구"]
```

쉼표 제거하고 문자형 데이터를 수치형 데이터로 변환 (3)

```
def    delcomma(col) :
       temp = []
       for i in col :
            a = i.replace(',','')
            a = float(a)
            temp.append(a)
       return temp
```

반복문으로 전체칼럼에 함수 적용 (4)

```
for colName    in colList:
      df[colName] = delcomma(df[colName])
```

상관계수 시각화 (5)

```
col       =   list(df.columns.difference(["행정구
역","지역구"]))
dfCorr    =   df[col].corr()
dfCorr
sns.heatmap(data=dfCorr,annot=True,
fmt='.2f', linewidths=.5, cmap='Blues')
```

정규화 (6)

```
x= df[col].values
min_max_scaler=preprocessing.MinMaxScale
r()

x_scaled=
min_max_scaler.fit_transform(x.astype(float))
df_nomal=pd.DataFrame(x_scaled,
                      columns=col,
                      index=df.index)

df_nomal.head()
```

지역구, 행정구역 다시 합치기 (7)

```
df_nomal[“지역구”] = df[‘지역구’]
df_momal[‘행정구역’] = df[‘행정구역’]
df_nomal.head()
```

지도에 매핑하기 (9)

```
data = pd.read_csv(“랜덤포레스트 예측결과.csv”, encoding=”euc-kr”)
data.head()

result = data[[“행정구역”, “Degree”]]
result.index = result[“행정구역”]

map = folium.Map(location = [37.5502, 126.982],
                 zoom_start=11)
map.choropleth(
    geo_data=geo_str,
    data=result[‘Degree’],
    columns=[result.index, result[‘Degree’]],
    fill_color=’YlGnBu’,
    key_on = ‘feature.properties.adm_dr_nm’
)
map
```

geojson 파일 경로 설정 (8)

```
geo_path=’/Users/User/Documents/이륜차/4326seoul.geojson’

geo_str=json.load(open(geo_path, encoding=’utf-8’)
```

다. 참고자료 & 참고문헌

참고자료

출처 1. [뉴스] 기획재정부 한국판 뉴딜 종합계획 발표
http://www.moef.go.kr/nw/nes/detailNesDtaView.do?searchBbsId=MOSFBBS_000000000028&menuNo=4010100&searchNttId=MOSF_000000000040637

출처 2. [뉴스] 코로나 19: 탄소 배출량 급감… 코로나 19 는 기후 위기를 멈출까?
https://www.bbc.com/korean/news-52540743

출처 3. [뉴스] “전기 오토바이로 배달”… 국토부, 삼성 SDI 등과 ‘그린배달 서포터즈’ 출범
http://www.greenpostkorea.co.kr/news/articleView.html?idxno=119276

출처 4. [뉴스] “4 년째 보조금 줬건만…” 전기오토바이의 씁쓸한 현주소
https://www.thescoop.co.kr/news/articleView.html?idxno=40373

출처 5. [뉴스] 배달 폭주에 오토바이 업종도 질주
https://www.sedaily.com/NewsVIew/1Z5BQ1EWPD

출처 6. [뉴스] 서울 정책 아카이브 전기이륜차 보급 정책
https://www.seoulsolution.kr/ko/content/1386

참고문헌

출처1. [논문] 이륜차의 일 주행거리조사와 대기오염 배출량 추정
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE01371289

출처2. [논문] 이륜차의 온실가스 배출량 추정
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE02411743