고립 사고 방지를 위한 방법 고찰

i

최근 어린아이가 차량에 방치되어 안타깝게 목숨을 잃는 사고가 있었다. 단 한 번만이라도 확인했다면 일어나지 않았을 사고지만, 이러한 비극이 다시 일어나지 않으리라는 보장은 없다. 하지만 고의적으로 확인하지 않는 것이 아니라 확인하는 것을 잊어버리는 경우가 많으므로, 인간에게 맡기는 것은 확실한 해결방안이 될 수 없다. 따라서, 센서 등을 활용하여 자동으로 파악한 후 알려주는 것이 좀 더 나은 해결방안일 것이다.

그렇다면 무엇을 사용하여 이런 사고를 방지할 수 있을까. 열 감지 센서를 사용하여 사람의 유무를 파악할 수 있다면, 이 문제를 해결할 수도 있을 것이다. 하지만 예산소방서의 보도자료에 따르면, 여름철 차량의 실내 온도는 사람의 체온을 훨씬 넘는 90 도까지 올라가므로 열 감지 센서로는 적절한 해결방안이 되지 못한다.[ii] 따라서, 열 감지 센서를 보조 혹은 대체할 수단은 없을까 생각하던 중, 차량 내부 대기 및 환경의 상태에 따라 사람을 파악할 수 있도록 학습된 모델이 있다면 열 감지 센서와 더불어 이를 보완해줄 수 있지 않을까 생각했다.

UC Irvine 에서 차량 내부의 환경 정보는 아니지만, 그에 상응하는 데이터를 얻을 수 있었다. 'Occupancy Detection Data Set'이 그것이다. 과연 환경 정보에 따라 사람의 유무를 판별할 수 있을까? 이러한 불의의 사고가 다시 일어나지 않게 할 수 있을지 한번 확인해 보겠다.

분석 데이터 설명

분석에 쓰인 기초 데이터는 UC Irvine 에서 제공하는 'Occupancy Detection Data Set'을 사용하였다. 여기서 기초 데이터에 대한 설명을 간략히 하고, 후에 최종적으로 사용된 파생 및 추가 변수에 대해 설명하겠다.

1) 기본 데이터의 출처:

UC Irvine Occupancy Detection Data Set: *https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection+*

※ 본래 이 데이터셋은 빌딩의 효율적인 에너지 소모 연구를 위해 수집되었으며, 사무실 내부의 온도, 습도, CO2, 빛 정보들을 수집하고 사람의 유무를 측정한 것이다.

2) 변수 이름과 설명

	date	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
140	2015-02-02 14:19:00	23.7000	26.272	585.200000	749.200000	0.004764	1
141	2015-02-02 14:19:59	23.7180	26.290	578.400000	760.400000	0.004773	1
142	2015-02-02 14:21:00	23.7300	26.230	572.666667	769.666667	0.004765	1
143	2015-02-02 14:22:00	23.7225	26.125	493.750000	774.750000	0.004744	1
144	2015-02-02 14:23:00	23.7540	26.200	488.600000	779.000000	0.004767	1

〈OCCUPANCY DETECTION DATA 의 표본〉

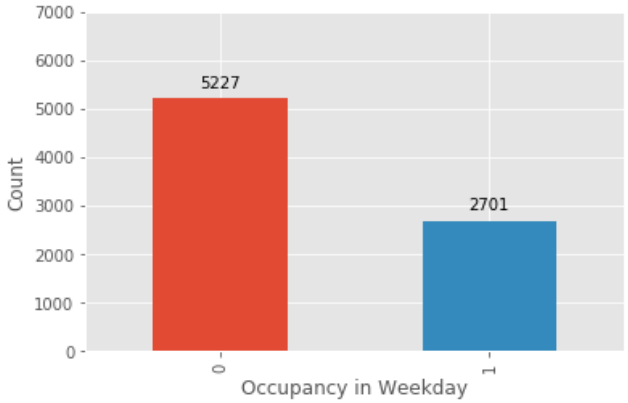
※ 총 20,560 개의 데이터로 구성됨

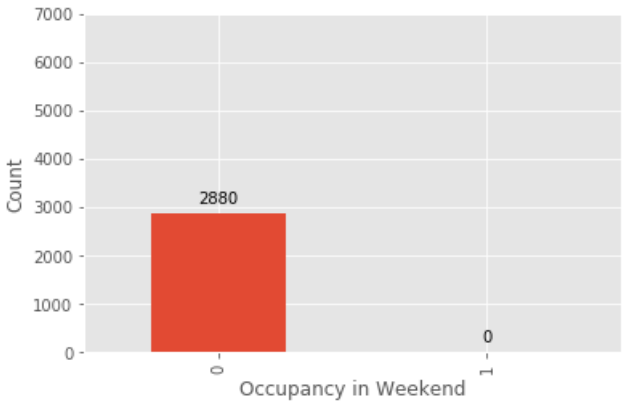
변수	설명
date	yyyy-mm-dd hh:mm 형식의 datetime 이다. 2015 년 2 월 2 일부터 2015 년 2 월 18 일 까지 대략 1 분 단위로 나누어져 있다.
Temperature	섭씨 온도(°C) 데이터 셋의 온도는 19℃~24.5℃의 범위 내의 값을 갖는다.
Humidity	상대 습도(%) 상대 습도란, 특정한 온도의 대기 중에 포함되어 있는 수증기의 양을 그 온도의 포화 수증기량으로 나눈 값이다.[iii] 데이터 셋의 습도(16% ~ 40%)는 실내에서 측정됐기 때문에 상대 습도가 높은 수치의 분포를 가지지는 않는다.
Light	빛(lux) 100 lux 는 매우 어두운 낮의 조명도이고, 1000 lux 는 일반 TV 스튜디오에서 쓰이는 밝기조명 정도의 조명도이다.[iv] 이상치를 제거하여 0 ~ 844lux 의 분포를 가질 수 있도록 조정했다.
CO2	대기 중의 이산화탄소 농도(ppm) ppm 이란 백만분의 1 이라는 의미이며, 예를 들어 750ppm 은 750/1000000*100 = 0.0750% 이다.
HumidityRatio	Temperature 와 Humidity 로부터 유도된 수치(데이터 제공자가 만든 파생 변수)이다.
Occupancy	사람의 유무 [0 - 없음, 1 - 있음]

EDA

[어떤 시간 대에 사람이 있을까?]

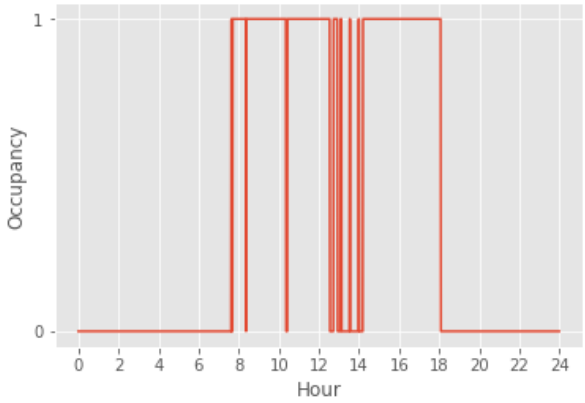
먼저, 주중과 주말로 나누어 사람이 있을 때와 사람이 없을 때 데이터 개수는 다음과 같다.





주말에는 사람이 항상 없었다. 아무래도 이 데이터가 수집된 장소가 사무실이다 보니 주말에 사람이 있지 않을 것이라고 추측했다. 주중에도 사람이 있을 때보다 사람이 없을 때가 더 많다.

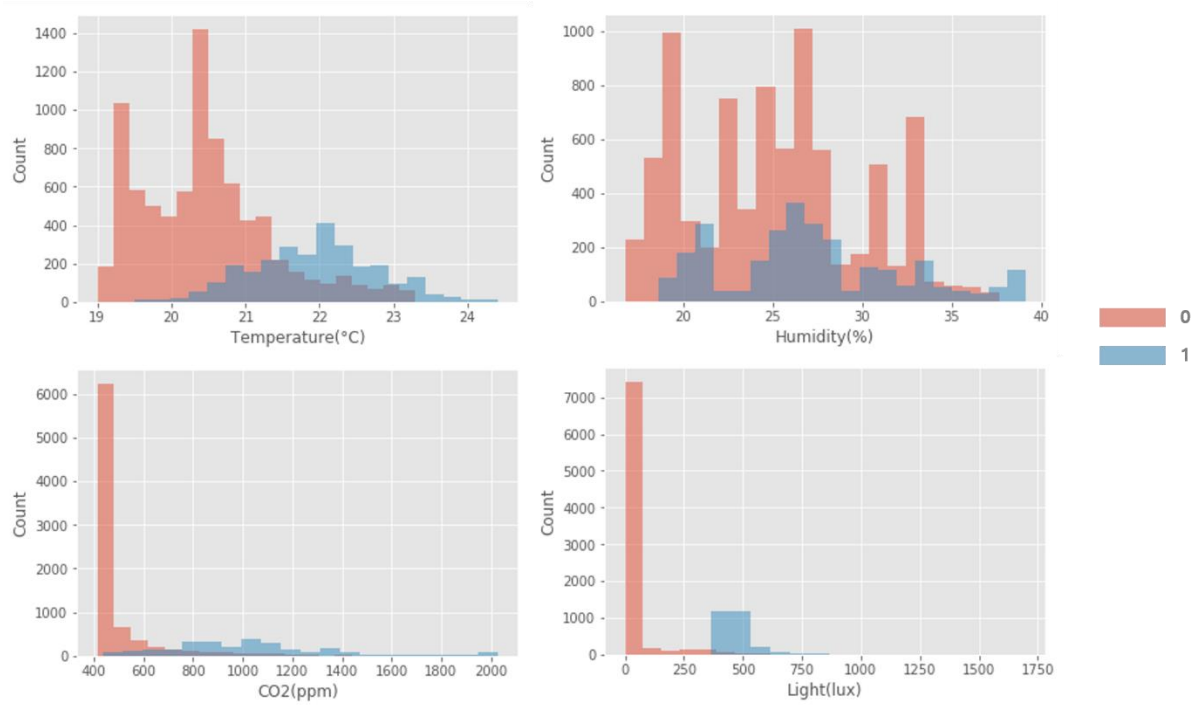
사람이 머무르는 주중에 수집된 데이터를 살펴보자. 일부 표본 데이터를 이용하여 주로 언제 사람이 있는지 그리고 언제 사람이 자주 들락날락하는지 알아보았다. 다음 그림은 주중인 2 월 5 일에 시간에 따른 사람의 유무를 나타낸다.



사람들은 대략 오전 8 시부터 오후 6 시까지 사무실에 머무른다. 출근시간이라 추정되는 오전 8 시 전후와 퇴근시간이라 추정되는 오후 6 시에 사람의 유무가 자주 변한다. 특히, 점심시간이라고 추정되는 오후 12 시부터 2 시 사이에 사람의 유무가 크게 변한다.

[사람이 있을 때와 없을 때 온도, 습도, CO2, 빛의 값의 차이가 있을까?]

사람의 유무와 환경 정보(온도, 습도, CO2, 빛)간의 관계를 파악하기 위해서 python 의 pyplot 을 이용하여 히스토그램을 그려보았다. 사람의 유무에 따라 두 집단의 히스토그램을 다른 색으로 표현하였다. *Temperature*, *Humidity*, *CO2*, *Light* 의 분포를 그리면 다음과 같다.



사람의 유무(*Occupancy*)에 따른 데이터의 개수는 다음과 같다

- 사람이 존재(1) : 8,107 개
- 사람이 존재하지 않음(0) : 2,701 개

사람이 있을 때의 데이터가 그렇지 않을 때의 데이터보다 약 3 배 많기 때문에 이 점을 고려하여 분포를 살펴 보았다.

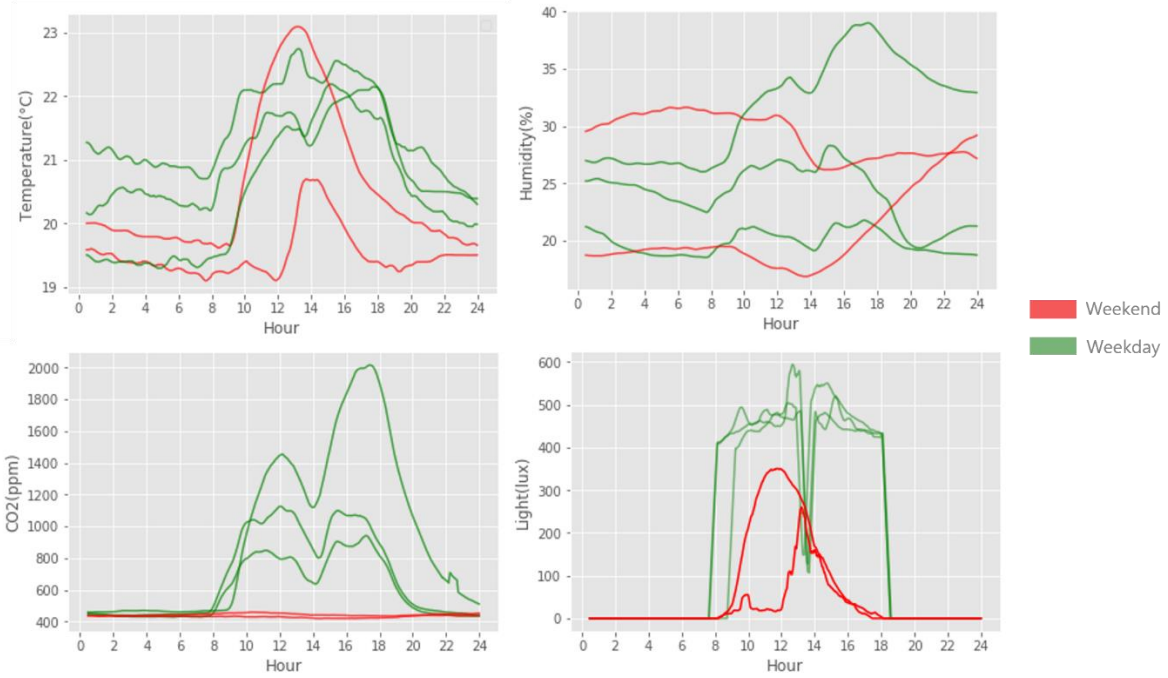
분포의 겹친 정도는 각각 다르지만 모든 분포에 대해 사람이 존재할 때 환경 정보 분포들이 그렇지 않을 때의 분포들보다 더 오른쪽에 위치해 있다. 즉, 사람이 존재할 때 환경 정보 측정값의 최솟값과 최댓값이 그렇지 않을 때의 값들 보다 더 크다고 할 수 있다. 결론적으로 사람이 있을 때 환경 정보 측정값들은 대체로 더 높은 값을 가진다.

Humidity 의 분포는 꽤 많은 부분이 겹쳐 있는 반면에, *Temperature* 의 분포는 상대적으로 분리되어 있다.

눈에 띄는 점은 *Light* 와 *CO2* 의 분포인데, 두 분포 모두 두 집단 사이의 구분이 훨씬 분명한 편이다. Light 와 CO2 가 *Humidity* 보다는 사람의 유무를 구분하는 기준으로 적합하다고 할 수 있다. 하지만 앞서 살펴보았듯이 주로 밤 시간 대에는 사람이 없었고 그 특징들이 *Light* 의 분포에 영향을 미쳤을 가능성이 있다. 왜냐하면 사람이 없을 때 대부분의 *Light* 값들은 0~20lux 사이에 분포해 있고, 이 밝기는 밤 하늘의 밝기이기 때문이다.

[사람이 없어도 시간에 따라 온도, 습도, CO2, 빛의 값이 변할까?]

온도는 사람의 유무뿐 아니라 시간 변화에 따라서도 달라진다. 새벽에는 온도가 낮다가 낮이 되면서 온도가 증가하고 다시 밤이 되면 온도가 낮아진다. 이처럼 사람의 유무에 관계없이 환경 정보 측정값은 변할 수 있다. 주말(사람이 전혀 없는 날)과 주중(사람이 있는 날)의 시간에 따른 환경 정보 측정값의 변화를 비교하여 사람이 없는 자연 상태에서의 환경 정보의 변화를 알아보자. 시간에 따른 *Temperature*, *Humidity*, *CO2*, *Light* 의 분포를 그리면 다음과 같다.



Temperature 의 분포를 보면, 주말과 주중의 온도 차이가 크지 않다. 주중보다 주말의 최고 온도가 더 높기도 하다. 다만 주말에는 낮 시간 대 작은 온도 변동이 거의 없고 온도 변화가 훨씬 부드럽다는 패턴을 확인할 수 있다. *Light* 는 주말과 주중에 빛의 세기 차이가 크긴 하지만, 세기 변화는 *Temperature* 와 비슷한 패턴을 보인다.

또한 눈에 띄는 점은 *CO2* 와 두 그룹의 값 범위뿐 아니라 분포 패턴에서 확연한 차이가 난다. 주말 분포의 경우 *CO2* 의 변동이 거의 없이 400~500ppm 을 유지하는 반면, 주중 분포에서는 점심시간 직후로 추정되는 오후 2 시쯤 *CO2* 가 약간 감소하는 패턴이 보인다. 즉 *CO2* 는 사람의 유무를 구분하는 기준으로 아주 적합해 보인다.

Humidity 의 분포는 앞서 확인한 바와 같이 사람의 유무에 따라 특징적인 패턴이 보이지 않아 사람의 유무를 구분하는 기준으로 적합하지 않을 것이라고 판단된다.

가설 설정 및 파생 변수 생성

EDA 를 바탕으로 우리는 몇 가지 가설을 세우고, 파생 변수를 생성하였다.

[절대적인 온도 값 보다는 시간에 온도의 변화량이 사람의 유무를 구분하는데 영향을 미친다.]

다음은 표본 중 하루인 2 월 5 일에 시간에 따른 온도 변화와 사람의 유무를 보여준다.

사람이 들어왔을 때 혹은 나갔을 때 온도 변화가 일어난다는 사실을 알 수 있다. 그러므로 사람의 유무를 구분하는데 영향을 미치는 것은 절대적인 온도 값이 아니라 온도의 변화량이라고 생각하여 다음 세 가지 파생 변수를 생성하였다.

변수	설명
TemperatureDif	1 분 전 온도와의 차
TemperatureBeforeAvg	이전 20 분 동안 온도의 평균
TemperatureAfterAvg	이후 20 분 동안 온도의 평균

온도뿐 아니라 습도와 CO2 또한 시간에 따라 연속적으로 변하는 값이기 때문에 같은 가설에 따라 파생 변수를 생성하였다.

[해당 시각이 특정 시간대에 속한다면 사람의 유무를 구분하는데 영향을 미친다.]

분석 데이터는 사무실에서 수집된 데이터이며 사무실에는 출·퇴근 시간, 근무시간, 점심시간 등 실제로 의미가 부여된 시간 구간이 많다. (Ex.주말에는 근무시간이 아니므로 사람이 없다.) 따라서 특정 시간대가 사람의 유무를 구분할 수 있다고 생각하여 다음과 같은 세 가지 파생 변수를 생성하였다.

변수	설명
weekday	주말(0), 주중(1)
office	비근무시간(0), 근무시간(1)
timegroup_(0~4)	새벽(0~6)(0), 오전(6~12)(1), 오후(12~18)(2), 밤(18~24)(3)

후에, 이 데이터 분석을 서론에서 제시한 것과 같이 안전 분야에 적용하기 위해서는 밤이나 새벽과 같은 사람이 없다고 기대되는 시간대에, 이 모델이 사람의 존재 여부를 더 예민하게 탐지하도록 해야 한다. 따라서 이 파생 변수는 후에 misclassification cost matrix 를 시간 구간에 따라 다르게 만들기 위해 사용될 것이라고 기대한다.

[외부 온도와 내부 온도의 차이가 사람의 유무를 구분하는데 영향을 미친다.]

실외 온도가 아주 높거나 낮을 때 내부에 사람이 있다면 실내외 온도차가 클 것이다. 따라서 실내외 온도 차로 사람의 유무를 구분할 수 있다고 생각하여 다음과 같은 파생 변수를 생성하였다.

변수	설명
Temp_InOut	실내외 온도 차

※ 최종 데이터

기본 데이터셋과 우리가 세운 가설을 통해 나온 데이터를 추가해 데이터 분석을 진행하려 한다.

1) 추가 변수

초반부에 나온 사고뉴스의 환경처럼 외부 온도에 대한 영향을 관찰하는 것도 중요하다고 판단하여, 추가 변수로 https://www.timeanddate.com/weather/belgium/mons/historic 에서 제공 되는 현지의 '외부 온도'와 '외부 습도'를 추가하였다.

2) 파생 변수

EDA 를 바탕으로 필요하다고 생각되는 파생 변수를 추가시켰다.

3) 분석에 사용된 최종 데이터

종속 ● / 기본 ● / 추가 ● / 파생 ●

변수	설명
Occupancy	종속 변수
date	기본 변수: 위와 동일
Temperature	
Humidity	
Light	
CO2	
HumidityRatio	
Out_Temp	추가 변수: 관측된 현지의 실제 외부 온도와 습도. 벨기에 몽스 지역에서 실험이 진행되었다는 사실을 알게 되어 직접 수집
Out_Humi	
TemperatureDif	파생 변수: 1 분전 각 측정값과의 차
HumidityDif	
CO2Dif	
HumidityRatioDif	
weekday	파생 변수:주말이면 0, 주중이면 1 로 category 변수를 생성
office	파생 변수: 07 시 ~ 19 시를 office hour 로 설정하여, office hour 내에 속하면 1, 아니면 0 으로 설정하여 category 변수를 생성
timegroup	파생 변수: 하루 24 시간을 6 시간씩 나누어, 더미 변수로 사용
TemperatureBeforeAvg	파생 변수: 이전 20 분 동안 측정값의 평균
HumidityBeforeAvg	
CO2BeforeAvg	
HumidityRatioBeforeAvg	
Temp_InOut	파생 변수: 실내 온도와 외부 온도와의 차
TemperatureAfterAvg	파생 변수: 이후 20 분 동안 측정값의 평균
HumidityAfterAvg	
CO2AfterAvg	
HumidityRatioAfterAvg	

자, 그럼 이제 모델링을 통해 확인해보자.

모델 비교 및 성능 평가

[변수 조합 및 선택]

우리가 모델을 학습시킬 때는 가설에 따라 변수를 각각 선택하여야 한다. 변수들의 종류는 다음과 같다.

첫 번째 가설 - 연속형 변수의 영향 차이(Dif, BeforeAvg, AfterAvg)
두 번째 가설 - 시계열 변수의 영향 차이(weekday, office, timegroup)
세 번째 가설 - 외부 온도 변수의 영향 차이(Temp_InOut, Temperature)

이에 따라, 총 18 가지 경우가 도출되며 명명은 다음과 같다.

Temp_InOut	**Dif**	**BeforeAvg**	**AfterAvg**
Weekday	wdt	wit	wat
Office	odt	oit	oat
Timegroup	tdt	tit	tat

Temperature	**Dif**	**BeforeAvg**	**AfterAvg**
Weekday	wdr	wir	war
Office	odr	oir	oar
Timegroup	tdr	tir	tar

[학습모델 생성]

우리가 사용할 모델의 종류는 다음과 같다.

a. Random Forest Classifier
모델 학습을 위해 Scikit-learn 패키지의 RandomForestClassifier 를 사용하였으며, Tuning 을 통해 사용한 인자는 다음과 같다.
Max_depth = 5
N_estimators = 150
Min_samples_leaf = 10
Oob_score = True
투입 변수 개수만큼, 총 18 개의 모델을 학습하였다.

b. Logistic Regression
모델 학습을 위해 Scikit-learn 패키지의 LogisticRegression 를 사용하였으며, 인자는 다음과 같다.
Penalty = l2
투입 변수 개수만큼, 총 18 개의 모델을 학습하였다.

[성능 평가 및 비교]

a. 표 설명
① 열에 있는 Test1_Accuracy 와 Test1_F1 Score 은 Train 된 모델을 첫 번째 테스트셋에 적용하여 각각 Accuracy score 와 F1 score 를 측정한 것이다.
② 열에 있는 Test2_Accuracy 와 Test2_F1 Score 은 Train 된 모델을 두 번째 테스트셋에 적용하여 각각 Accuracy score 와 F1 score 를 측정한 것이다.
③ Random Forest Classifier 모델의 열에 있는 OOB Score 는 Out-of-bag score 를 측정한 것이다.
④ 행에 있는 Original 은 기존 데이터 셋을 Training 에 이용하였다.
⑤ 행에 있는 oar 등과 같은 index 들은 18 개의 경우의 수를 알파벳으로 간소화한 것이다.

b. 기존 데이터셋의 성능 평가
1) Random Forest Classifier

Index	Test1_Accuracy	Test1_F1 Score	Test2_Accuracy	Test2_F1 Score	OOB Score
Original	0.96848	0.95732	0.98554	0.96538	0.98907

2) Logistic Regression

Index	Test1_Accuracy	Test1_F1 Score	Test2_Accuracy	Test2_F1 Score
origin	0.97824	0.97094	0.99292	0.98335

c. 변환된 데이터셋의 성능 평가

1) Random Forest Classifier[v]

Index	Test1_Accuracy	Test1_F1 Score	Test2_Accuracy	Test2_F1 Score	OOB Score
odr	0.97829	0.97023	0.99197	0.98096	0.98865
odt	0.97867	0.97077	0.99176	0.98048	0.98877
tdr	0.97676	0.96795	0.99320	0.98384	0.98815
tdt	0.97524	0.96577	0.99320	0.98384	0.98852
wdr	0.97790	0.96967	0.99300	0.98336	0.98865
wdt	0.97829	0.97027	0.99290	0.98312	0.98815

2) Logistic Regression[vi]

Index	Test1_Accuracy	Test1_F1 Score	Test2_Accuracy	Test2_F1 Score
odr	0.97867	0.97077	0.96983	0.92376
odt	0.97867	0.97077	0.96983	0.92376
oir	0.97714	0.96865	0.98208	0.95657
oit	0.97714	0.96865	0.98208	0.95657
wir	0.97676	0.96808	0.97724	0.94437
wit	0.97676	0.96808	0.97724	0.94437

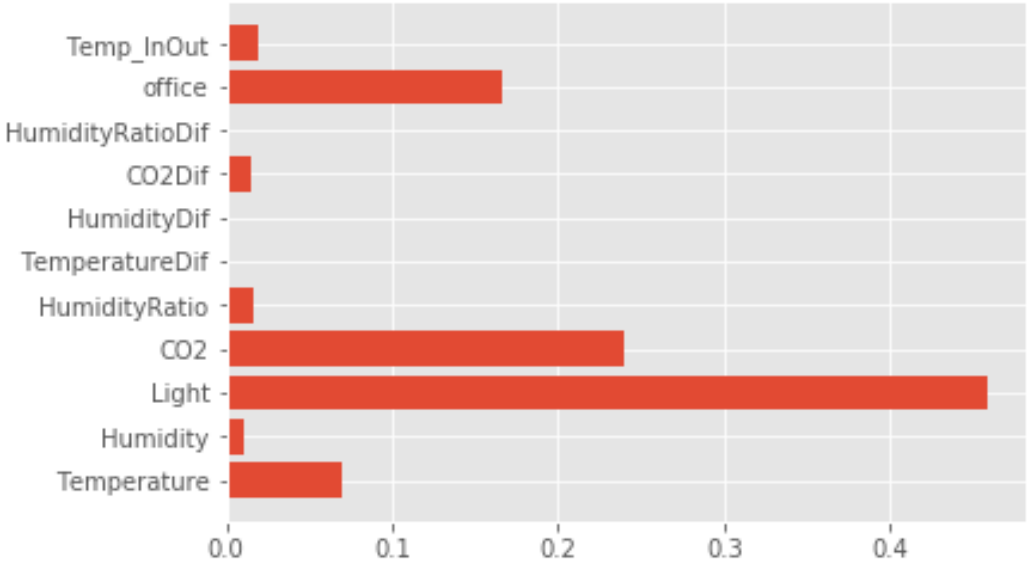
위 표는 변환된 데이터셋의 성능 평가 결과, 총 18 개 모델 중 점수가 높은 상위 6 개의 모델들의 점수이다.

[결론]

두 알고리즘의 상위 6 개의 모델에 odt 와 odr 이 공통적으로 포함되며, odt 변수 조합이 odr 변수 조합보다 조금 더 나은 성능을 보여준다.

변수 조합	연속형 변수	시계열 변수	외부 온도 변수
ODT	TemperatureDif HumidityDif CO2Dif HumidityRatioDif	Office	Temp_InOut

가장 높은 ODT 변수 조합을 학습한 Random Forest Classifier 의 importance 를 확인해보았다.



그 결과, *Light* 와 *CO2*, ***Office***, *Temperature* 순으로 중요도가 평가되었다. 또한, 파생 변수(*Temp_InOut*, *Office*, *CO2Dif*)들이 어느 정도의 중요도를 가지고 있는 걸로 보아, 성능 향상에 영향을 미쳤음을 알 수 있다.

Outro

지금까지 우리는 실내 환경에 따라 사람의 유무를 판단할 수 있는지 확인해 보았다. 다양한 시도를 통해서 우리가 세운 가설을 검증해보고, 결과적으로 유의미한 결과를 얻을 수 있었다. 하지만 아쉬운 점이 없는 것은 아니며, 몇 가지 한계점이 있었다.

- 한계점

1) 본 데이터셋 내의 특징들은 특정 사무실 환경에 국한되어 있기 때문에 실제로 적용할 때에는 그 환경에 맞는 데이터 수집과 새로운 모델링이 요구된다. 예를 들어, 한국과 벨기에의 날씨는 분명 다르며 다른 여러 수치도 측정되는 환경에 맞게 조정해줄 필요가 있다.

2) 또한, 고립 사고는 내부에서만 일어나는 것이 아니라 외부에서도 빈번하게 일어난다. 하천이나 산에서 불의의 사고로 인해 고립되는 사고에 대한 뉴스를 우리는 종종 볼 수 있다. 외부까지 확장하기에 센서 설치 문제도 있고, 그 범위를 설정 하는 것도 어려울 것이라는 한계점은 분명 존재한다.

그럼에도 불구하고, 이러한 환경 상태를 이용한다면 분명 실내 고립 사고는 예방될 수 있다.

- 활용방안

1) 냉동창고 혹은 해상 컨테이너 고립 사고 예방

2) 어린이 및 반려동물의 고립 사고 예방

3) 화재 시, 그 직전까지 미리 수집해 두었던 데이터를 통해 고립되어 있는 사람의 위치를 빠르게 예측하여 효율적인 구조 동선을 생성할 수 있다.


[i] http://ccnews.lawissue.co.kr/view.php?ud=20180719040253969 66a28b45db0_12

[ii] http://www.anewsa.com/detail.php?number=1349095&thread=09r02 (예산소방서, 폭염 속 차량 화재주의보)

[iii] https://en.wikipedia.org/wiki/Humidity

[iv] https://ko.wikipedia.org/wiki/럭스

ᵛ RandomForest 결과표

Index	Test1_Accuracy	Test1_F1 Score	Test2_Accuracy	Test2_F1 Score	OOB Score
oar	0.97676	0.96818	0.98538	0.96590	0.98902
oat	0.97790	0.96976	0.97941	0.95220	0.98902
odr	0.97829	0.97023	0.99197	0.98096	0.98865
odt	0.97867	0.97077	0.99176	0.98048	0.98877
oir	0.97752	0.96919	0.98558	0.96636	0.98865
oit	0.97790	0.96976	0.97498	0.94022	0.98889
tar	0.97257	0.96215	0.97941	0.95256	0.98778
tat	0.97790	0.96967	0.97035	0.92955	0.98828
tdr	0.97676	0.96795	0.99320	0.98384	0.98815
tdt	0.97524	0.96577	0.99320	0.98384	0.98852
tir	0.97562	0.96639	0.96314	0.90904	0.98741
tit	0.97333	0.96312	0.98085	0.95529	0.98704
war	0.97829	0.97020	0.98641	0.96821	0.98877
wat	0.97676	0.96818	0.98054	0.95514	0.98914
wdr	0.97790	0.96967	0.99300	0.98336	0.98865
wdt	0.97829	0.97027	0.99290	0.98312	0.98815
wir	0.97562	0.96649	0.97168	0.93001	0.98828
wit	0.97905	0.97122	0.98044	0.95187	0.98840

[vi] Logistic Regression 결과표

Index	Test1_Accuracy	Test1_F1 Score	Test2_Accuracy	Test2_F1 Score
oar	0.97295	0.96237	0.89374	0.77097
oat	0.97295	0.96237	0.89374	0.77097
odr	0.97867	0.97077	0.96983	0.92376
odt	0.97867	0.97077	0.96983	0.92376
oir	0.97714	0.96865	0.98208	0.95657
oit	0.97714	0.96865	0.98208	0.95657
tar	0.96914	0.95648	0.86501	0.70493
tat	0.96914	0.95648	0.86501	0.70493
tdr	0.97829	0.97023	0.95778	0.89037
tdt	0.97829	0.97023	0.95778	0.89037
tir	0.97181	0.96081	0.95150	0.87369
tit	0.97181	0.96081	0.95150	0.87369
war	0.97562	0.96621	0.88581	0.75959
wat	0.97562	0.96621	0.88581	0.75959
wdr	0.97829	0.97023	0.96386	0.90727
wdt	0.97829	0.97023	0.96386	0.90727
wir	0.97676	0.96808	0.97724	0.94437
wit	0.97676	0.96808	0.97724	0.94437