

# 머신러닝을 활용한 비접촉 온도 센서의 측정 온도 보정 계산 및 자동화

안양대학교

스마트창의융합대학

컴퓨터공학전공

서호준

지 도 교 수 박 성 순

이 논문을 공학 학사 학위논문으로 제출함.

2021년 12월

안양대학교

스마트창의융합대학

컴퓨터공학전공

서호준

# 논 문 인 준 서

이 논문을 서호준의 공학 학사 학위논문으로 인준함.

2021년 12월

주 심 박 성 순 인

부 심 김 성 규 인

부 심 인

# 차 례

요약 .....	1
제 1장 서론 .....	2
제 1절 연구 배경과 목적 .....	2
제 2절 연구 내용 .....	3
제 3절 연구 범위 한정 .....	5
 제 2장 보정 온도 데이터 생성 .....	6
제 1절 학습 데이터 저장 .....	6
제 2절 보정 값 계산 .....	7
제 3절 보정 값 평가 .....	9
 제 3장 보정 온도 관리 .....	10
제 1절 보정 온도 관리 .....	10
 제 4장 구현 및 테스트 .....	13
제 1절 보정 온도 계산 .....	13
제 2절 서버 구현 .....	14
제 3절 자동화 .....	16
제 4절 테스트 .....	17
제 5절 테스트 결과 .....	18
 제 5장 결론 .....	21
제 1절 문제점 .....	21
제 2절 결론 .....	21
[참고 문헌] .....	22

# 그림 차례

## 그림 차례

[그림 1] 기상청 날씨 누리에서 크롤링할 데이터 .....	6
[그림 2] K- 최근접 알고리즘 .....	7
[그림 3] 선형 회귀 분석 .....	7
[그림 4] 서버 순서도 .....	10
[그림 5] 서버 구성도 .....	11
[그림 6] 열 측정 기기 순서도 .....	12
[그림 7] 학습 데이터 테이블 .....	13
[그림 8] REST API 구현 .....	14
[그림 9] 갱신되는 데이터 .....	16
[그림 10] 구현한 비접촉 체온 측정기기 .....	17
[그림 11] 보정 온도의 정확도 .....	18
[그림 12] 보정 온도와 야외 온도 .....	18
[그림 13] 보정 온도와 야외 풍속 .....	19
[그림 14] 보정 온도와 실내 온도 .....	19
[그림 15] 보정 온도와 실내 습도 .....	20
[그림 16] 보정 온도와 야외 강수량 .....	20
[그림 17] 보정 온도와 야외 습도 .....	20

## 요 약

비 접촉 온도 센서의 단점인 측정 온도의 정확도가 떨어지는 것을 보정 온도를 계산 후 합산하여 보정하고자 한다. 보정 온도는 바깥 온도와 내부 온도를 통해 별도의 서버 컴퓨터에서 머신러닝을 통해 계산한다. 그 후 인터넷을 통해 데이터를 배포한 후 체온 측정 기기에서 데이터를 가져와서 사용하기까지의 과정을 보여준다. 또한 데이터를 계산한 결과들을 분석하여 어떠한 환경 조건이 비접촉 체온 측정에 영향을 미치는지 확인을 할 것이다.

# 제 1장 서 론

## 제 1절 연구 배경과 목적

현재 이 논문을 작성하는 시점에서도 아직 코로나가 종식되지 않고 아직 확산되고 있는 추세이다. 우리나라에 첫 확진 환자가 발생한 지 2년이 다가가는 지금에도 일일 확진자는 줄어들 추세를 보이지 않는다. 또한 백신이 완전히 보급되는 데에 시간이 소요되며 백신을 맞고도 코로나에 걸리는 돌파 감염의 가능성, 코로나 외 델타, 뮤 등 다른 변종 등 수많은 위험 요소들이 있어 이러한 상황이 종식될 기미가 보이지 않는다.

치료제 또는 백신을 개발하기 위해서는 긴 시간이 필요한데 이 시간 동안에 최대한 바이러스의 전파를 늦추는 것이 중요하다. 기업, 기관, 매장 등 많은 사람이 출입하는 곳은 체온을 측정하여 고온인 사람의 출입을 배제하여 혹시 모를 감염을 예방하고 출입자의 신원을 미리 파악하여 감염 확산을 막기 위한 조치를 빠르게 행할 수 있다.

온도를 측정하는 기기는 비접촉 센서를 사용하는 것이 대부분이다. 사람의 피부에 접촉하는 면이 없기 때문에 온도 측정 시 접촉을 통한 감염을 최소화할 수 있고 빠른 시간에 온도를 측정하기 때문에 이 센서를 많이 사용하고 있다.

하지만 비접촉 센서의 측정 온도는 정확도가 낮다는 것이 단점이다. 특히 사람의 평균 체온은 36.3도 내외 인데 겨울에 측정하면 대부분의 사람들이 35.5도 내외로 측정되는 것을 볼 수 있으며 심지어 체온이 너무 아래로 측정되어 기기가 측정이 잘못되었다면서 다시 측정해야 하는 경우도 있다.

비접촉 센서가 접촉 센서에 비해 차이가 나는 이유는 비접촉 센서가 열을 측정하는 방식에서 발생한다. 비접촉 센서는 가시광선이 아닌 사람이 발출하는 적외선을 받아들여 그것을 디지털 또는 아날로그로 표현한다. 여기서 사람의 표면과 카메라 사이에 있는 물질에 따라 측정값이 달라진다. 공기에 있는 수분이나 직사광선 등 자잘한 요소들이 영향을 준다는 것이다.

그래서 이러한 요소들의 영향들을 최소화하기 위해 주변 환경에 따라 측정 온도에 보정 온도를 계산 후 합산하여 측정 온도에 대한 신뢰성을 증가시키고자 한다.

## 제 2절 연구 내용

비접촉 온도 측정에는 여러 요소들이 영향을 받는다.

### 1. 열 전도성

재료마다 열 전도성이 다르다. 예를 들어 절연체는 천천히 가열되는 반면 금속을 빠르게 가열된다. 이 차이는 특정 상황에서 큰 온도차를 일으킬 수 있다.

### 2. 복사율

복사율은 물체가 적외선을 방출하는 효율이다. 이것 또한 물질의 재료에 따라 다르다.

### 3. 반사도

일부 재료는 거울과 같이 열 방출을 반사한다. 대표적인 것이 연마된 비산화 금속이 있다. 반사로 인해 열 화상 이미지를 잘못 해석할 수 있으므로 카메라의 각도를 세부적으로 조정할 수 있어야 한다.

### 4. 주변 환경

주변 온도는 온도 검사에 큰 영향을 미친다. 직사광선에 노출되면 더 이상 노출이 되지 않아도 몇 시간 동안 표면의 열에 영향을 미친다. 또한 바람과 비는 표면을 냉각시킬 수 있으며 비 같은 경우 표면의 수분이 증발되면서도 냉각이 된다.

여러 요소 중에서 주변 환경에 대한 요소에 대해 연구하는 것이 좋을 것이라 생각하였다. 열 전도성 및 복사율은 사람의 피부를 대상으로 측정하기 때문에 재료에 따라 측정하는데 문제가 발생하지 않는다. 반사도로 인한 열 감지 문제는 발생할 수 있다. 뒷면의 거울이나 유리가 위치함에 따라 온도를 측정하는데 오류를 발생시킬 수 있다. 하지만 이것은 거울의 방향을 바꾸거나 커튼을 사용하는 등 쉽게 해결할 수 있다.

그럼 주변 환경은 크게 외부와 내부로 나눌 수 있다. 외부의 환경은 날씨에 의해 좌우가 되는데 기온이 높으면 그만큼 직사광선이 강해지고 직사광선을 쬔 피부의 표면 온도는 증가하게 된다. 그리고 바람이 강하면 표면이 냉각되며 비가 오면 표면의 물기가 증발하면서 냉각이 된다.

내부 환경은 온도를 측정하는 장소의 환경이다. 측정하는 장소의 환경 상태에 따라 측정하는 값이 달라진다. 더운 여름에 온도를 측정할 때 냉방을 강하게 한 매장



과 냉방을 하지 않는 매장을 같이 비교하면 데이터가 달라서 오류가 발생한다. 온도를 측정할 때 내부 출입 시간과 측정 시간에 약간의 시간이 소요되는데 이 때 피부의 땀 등이 증발하면서 피부가 냉각되거나 히터 등으로 피부가 가열되어 측정 온도가 달라 질 수 있다.

이러한 문제 때문에 주변 환경의 데이터를 기반으로 보정 온도를 계산하여 측정 온도와 합산하여 정확한 측정 온도를 구하고자 한다. 온도, 습도, 직사광선, 강수량 등 주변 환경의 데이터와 해당 환경에 있는 사람의 비접촉 센서의 측정 온도와 접촉 센서의 측정 온도를 구한다. 그 후 접촉 센서의 측정 온도에서 비접촉 센서의 측정 온도의 차로 보정 온도를 구한 후 이것을 학습 데이터로 머신 러닝을 통해 학습시킨다.

### 제 3절 연구 범위 한정

해당 연구에서 특수한 환경 또는 행동을 한 사람들의 체온 측정은 배제한다. 가령 찜질방에 있다가 식당에 바로 들어오는 경우나 운동하고 온 후 씻지 않고 바로 식당에 오는 경우 등 이다. 이러한 경우는 주변 환경보다 다른 행위 또는 특수한 환경의 영향을 많이 받기 때문이다. 본 논문에서는 주변 환경에 따른 체온 측정을 보정하기 위한 것이기 때문에 주변 환경에 영향을 적게 받은 사람의 체온을 측정하는 것은 어렵다.

체온 측정기기는 반드시 유선 또는 무선으로 인터넷에 연결되어 있어야 한다. 체온 측정기기가 내부의 센서와 바깥의 환경의 데이터를 이용하여 보정 온도를 구할 수 있다. 하지만 처리 속도 문제, 정보 보안 문제 등을 야기 할 수 있기 때문에 해당 논문에서는 인터넷을 사용하여 보정 온도를 가져오도록 하였다. 우선 체온 측정기기의 처리속도가 비용 등의 문제로 인해 처리 속도가 느릴 수 있고 보정 온도를 측정할 때 사용되는 학습데이터는 소중한 자산이므로 이것을 측정기기에 보관하면 다른 사람들이 악용 할 수 있어 별도의 서버에서 관리하는 것이 좋다.

## 제 2장 보정 온도 데이터 생성

### 제 1절 학습 데이터 저장

우선 외부 환경 데이터부터 저장한다. 데이터는 기상청 날씨 누리(<https://www.weather.go.kr/weather/observation/currentweather.jsp>)에서 크롤링을 통해 데이터를 가져온다. 학습 데이터를 만드는 장소의 가장 가까운 지점을 선택하고 선택한 지점의 데이터를 가져오는데 직사광선과 관련이 있는 기온, 표면의 물기와 관련 있는 강수와 습도, 표면의 냉각에 관련 있는 풍속을 가져온다.

지점	날씨				기온(°C)			강수		바람		기압(hPa)
	현재일기	시정 km	운량 1/10	중하운량	현재 기온	이슬점 온도	체감 온도	일강수 mm	습도 %	풍향	풍속 m/s	해면 기압
강릉		20 이상			24.9	15.7	25.6		57	동북동	1.9	1021.8
강진군		19.6			24.2	17.9	25.8	0.0	68	동남동	1.9	1016.6
강화		20 이상			25.6	15.5	26.1		54	남남동	0.5	1019.6
거제		20 이상			23.4	15.1	24.4		60	동	1.9	1019.2
거창		20 이상			22.6	14.6	23.7		61	동북동	1.7	1018.8
경주시		20 이상			21.9	15.4	23.4		67	북동	3.9	1020.3

[그림 1] 기상청 날씨 누리에서 크롤링할 데이터

내부 환경의 데이터는 온습도 센서를 통해 가져온다. 온도 센서를 통해 내부 온도에 의한 표면 온도의 냉각이나 가열을 확인할 수 있으며 습도 센서를 통해 표면의 수분이 증발함에 따라 냉각이 되는지 확인할 수 있다.

비접촉 온도 측정 센서는 2개로 나눈다. 하나는 손목 측정에 사용하는 MLX90614, 그리고 얼굴 온도 측정을 위한 flir lepton 2.5이다. MLX90614는 가장 많이 사용되는 온도 측정 센서이며 flir lepton 2.5는 신뢰성 있는 산업용 열 화상 카메라이다. 사실 의료용 열 화상 카메라가 정확도는 높지만 의료 기기라 별도의 허가 없이는 구매하기 힘들어 산업용 카메라로 대체할 수 밖에 없었다.

접촉 온도 측정은 귀에 꽂아서 측정하는 체온계를 선택하였다. 외부의 환경에 영향을 덜 받는 부분이며 접촉 체온 측정 방식 중에서 가장 빠르기 때문이다.

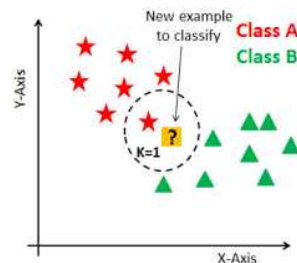
측정한 온도들을 통하여 보정 온도를 계산한다. 계산하는 방법은 (접촉 측정 온도) - (비 접촉 측정 온도) 로 한다. 구한 보정 온도는 주변 환경 데이터와 함께 학습 데이터로 사용되기 위해 저장된다.

## 제 2절 보정 온도 구하기

저장한 외부 환경 데이터와 내부 환경 데이터, 보정 온도들을 지도학습을 통해 학습시킨다. 보정 온도를 구할 수 있기 때문에 정답을 알고 있는 학습 데이터를 사용하는 지도학습을 택했으며 그 중에서 보정 온도라는 연속적인 숫자를 계산하기 위한 회귀 방식을 선택하였다. 그래서 3개의 회귀 알고리즘을 사용하여 보정 온도를 계산 후 비교하여 최적의 알고리즘을 구한다.

### 1. K- 최근접 이웃 알고리즘

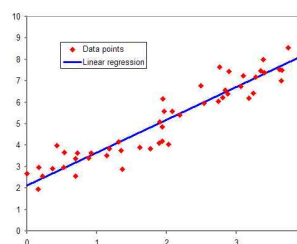
비교 대상이 되는 데이터 포인트 주변에 가장 가까이 존재하는 k개의 데이터와 비교해 가장 가까운 데이터 종류로 판별한다. 연속형 데이터 경우 k개의 데이터의 평균값으로 예측한다.



[그림 2] k- 최근접 이웃 알고리즘

### 2. 선형 회귀 분석

피쳐 데이터와 타겟 데이터 간의 선형 관계를 파악하는 알고리즘이다. 여기서의  $w$ 는 가중치라고 부르며 가중치 요소 하나하나가 파라미터이며 예측 값에 영향을 준다. 각 가중치는 기존 학습 데이터로부터 최소 제곱법을 사용해 구할 수 있다.



[그림 3] 선형 회귀 분석

### 3. 회귀 결정 트리

오차를 가장 잘 줄일 수 있는 변수를 기준으로 분기를 만들어 결과를 예측하는 모델이다. 변수가 중요한지 변수의 값에 따라 예측 결과가 무엇인지 한 눈에 볼 수 있어 설명력이 좋은 장점이 있다.

이러한 3개의 알고리즘을 비교하기 전에 먼저 고려해야하는 것이 있다. 먼저 차원 축소를 하여야 한다. 데이터에는 중요한 부분과 중요하지 않는 부분이 존재한다. 그 중 중요하지 않은 부분은 데이터에서 정보를 얻을 때 오류를 발생시키니 불필요한 부분을 제거하는 것이 중요하다. 추가적으로 학습 데이터의 차원이 커질수록 해당 차원을 표현하기 위해 필요한 데이터가 기하급수적으로 늘어나는 “차원의 저주”를 해결 할 수 있다.

두 번째로 정규화이다. 어느 한 데이터의 특성 스케일이 심하게 차이나면 그 데이터에 의해 예측 값이 좌지우지된다. 이러한 것을 방지하고자 모든 데이터가 동일한 스케일로 반영이 되도록 만드는 것이다.

차원 축소와 정규화를 거친 3개의 알고리즘으로 구현된 보정 온도들은 보정 온도 평가를 통해 가장 좋은 결과를 나타낸 알고리즘을 찾을 것이다.

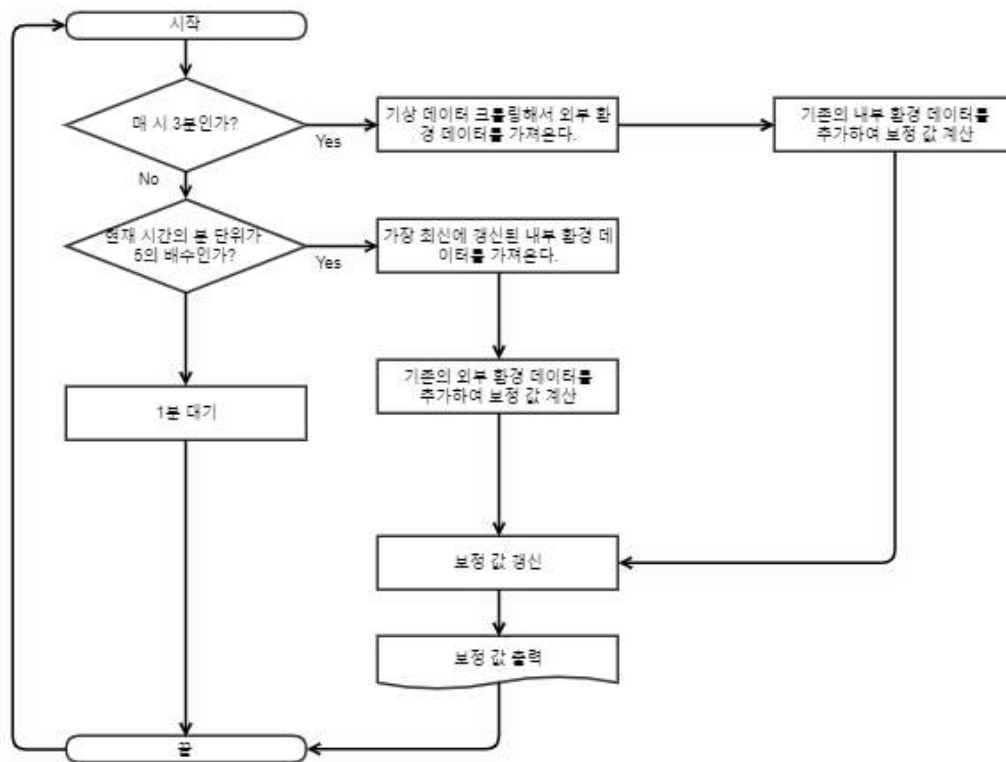
### 제 3절 보정 온도 평가

3개의 알고리즘으로 구현한 보정 온도들을 statsmodel 라이브러리를 사용하여 계수를 분석해 평가한다. 그 중에서 결정 계수 또는 수정 결정 계수라는 점수로 회귀 분석이 얼마나 잘 되었는지 평가할 수 있는데 이 점수들이 1에 가까울수록 데이터를 잘 설명하는 모델이라는 것을 알 수 있다.

## 제 3장 보정 온도 관리

### 제 1절 보정 온도 관리

#### 2.1.1. 서버



[그림 4] 서버 순서도

우선 학습 데이터는 별도의 서버에서 관리를 한다. 우수한 예측 모델을 구축하려면 우수한 데이터가 필요한데 중요한 데이터를 각각의 기기에 저장하는 것은 비효율적이며 악용이 될 소지가 있다.

내부 환경 데이터로 인한 갱신은 5분에 한 번씩 한다. 체온 측정기기 안에 내장된 컴퓨터에서 센서를 통해 온도, 습도를 측정하여 데이터베이스 서버에 데이터를 저장한다. 이렇게 시간마다 저장하는 이유는 체온 측정기기를 통해 온도를 측정할 때 빠른 결과를 얻기 위해서이다. 만약 이렇게 미리 측정하지 않는다면

1. 온도 측정
2. 주변 온도 측정
3. 데이터를 데이터베이스 서버에 전달

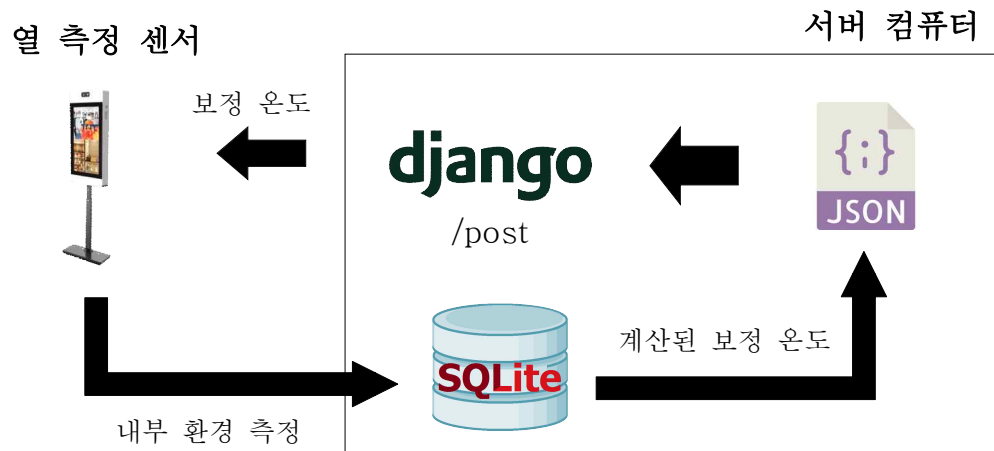
4. 데이터베이스 서버에 있는 데이터들을 통해 보정 온도를 구함
5. 보정 온도를 채온 측정기기에 전송
6. 보정 온도와 측정 온도를 합산 후 화면에 출력

이러한 순서가 될 것이다. 하지만 미리 측정을 하면

1. 온도 측정
2. 저장되어 있는 보정 온도에 측정 온도를 합산 후 화면에 출력

이렇게 간단하고 빠르게 출력할 수 있다.

외부 환경 데이터는 별도의 서버 컴퓨터에서 크롤링을 통해 데이터베이스에 저장한다. 날씨 데이터는 매 시 정각에 갱신이 되기 때문에 갱신 딜레이를 감안하여 매 시 3분마다 자동으로 외부 환경 데이터를 크롤링해서 저장 후 갱신한다.



[그림 5] 구성도

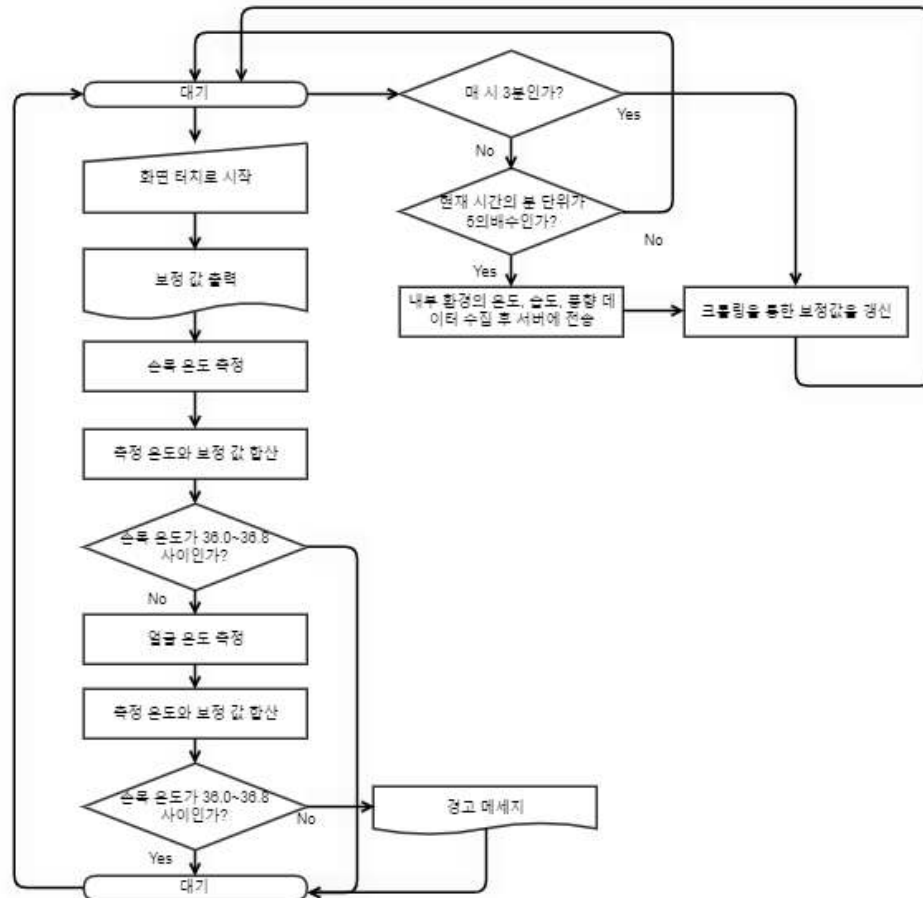
외부 환경 데이터와 내부 환경 데이터를 이용해서 보정 온도를 계산할 때마다 배포되는 보정 온도를 갱신한다. 서버 컴퓨터는 갱신한 데이터를 json 파일로 만들어 배포한다. 해당 파일은 측정기기의 고유 번호, 각 온도 측정 센서에 해당되는 보정 온도로 구성 되어 각 채온 측정기기의 환경에 맞는 보정 온도를 제공 할 수 있다.

배포된 보정 온도는 각 기기들이 크롤링을 통해 가져온다. 온도 측정 기기들은 자기 자신의 고유 번호를 찾아 해당되는 보정 온도를 가져오기 때문에 보정 온도를 직접 계산하는 부담이 줄어든다. 따라서 온도 측정 기기의 성능이 높을 필요가 없기 때문에 온도 측정 기기의 단가를 획기적으로 줄일 수 있다.



마지막으로 온도 측정 기기에서 내부 환경을 측정하고 그것을 서버의 데이터베이스에 접속하여 저장한다. 이 정보는 매 시간마다 서버에서 보정 온도를 계산하기 위한 데이터가 된다.

### 2.1.2. 체온 측정기기



[그림 6] 열 측정 기기 순서도

기본적으로 체온 측정기기는 대기 상태에서 데이터 갱신이 이루어진다. 작동 상태에서는 지정된 시간이 지나도 갱신이 되지 않으며 작동 후 대기 상태에 들어갔을 때 데이터 갱신을 한다. 매 시 3분 그리고 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55분마다 보정 온도를 크롤링을 통해 가져온다.

해당 데이터는 손목 온도 측정과 얼굴 온도 측정에 사용된다. 손목 측정 온도에 보정 온도를 합산하여 37.5도 이상이면 얼굴로 다시 측정한다. 그리고 얼굴 측정 온도와 보정 온도의 합마저 37.5도 이상이면 기기가 경고 메시지를 출력한다.

## 제 4장 구현 및 테스트

### 제 1절 보정 온도 계산

#### 3.1.1. 학습 데이터 저장

학습 데이터는 예측의 정확도를 평가하는 매우 중요한 데이터이다. 따라서 데이터를 얻을 때에는 특정된 한 공간에서 진행하는 것이 좋다. 공간마다 특수한 환경이 있어 예상치 못한 변수가 발생할 수 있기 때문이다. 그렇기 때문에 날씨 데이터는 서울을 기준으로 특정 장소는 한 곳에서만 진행하는 것이 좋다. 기상청에서 (<https://www.weather.go.kr/weather/observation/currentweather.jsp>)에서 크롤링을 통해 날씨 데이터를 가져온다.

내부 환경은 더욱더 한정된 조건에서 측정한다. 먼저 바깥 환경에 10분 이상~30분 미만 활동한 사람을 측정한다. 단시간 동안 밖에 있는 경우 외부 환경에 영향이 적으며 30분 이상 밖에서 활동한 사람은 소수일 것이라 판단하였기 때문이다.

데이터는 sqllite라는 데이터베이스에 저장한다. 해당 DB는 측정기기에 있는 것이 아닌 서버 컴퓨터에 위치한다. 속성 중 보정 온도만 null값을 가질 수 있다.

기온	일강수	습도	풍속	내부습도	내부온도	손목 보정온도	얼굴 보정온도

[그림 7] 학습데이터 테이블

#### 3.1.2. 보정 온도 계산하기

선형 회귀 분석을 사용하여 계산하였다. 사실 3개의 알고리즘을 사용하여 어떠한 것이 효율적인지 비교하려했지만 적은 학습 데이터 셋으로 인해 모든 모델들이 매우 높은 정답률을 가지고 있어 어떠한 알고리즘이 정확한지 비교할 수가 없었기 때문에 일단 현재 여러 번 재측정하여 가장 많이 우수한 결과를 보인 선형 회귀 분석을 채택하였다. 또한 여름 데이터가 추가된다면 현재 측정된 결과가 달라질 수 있기 때문에 추후에 결과가 달라질 수 있다.

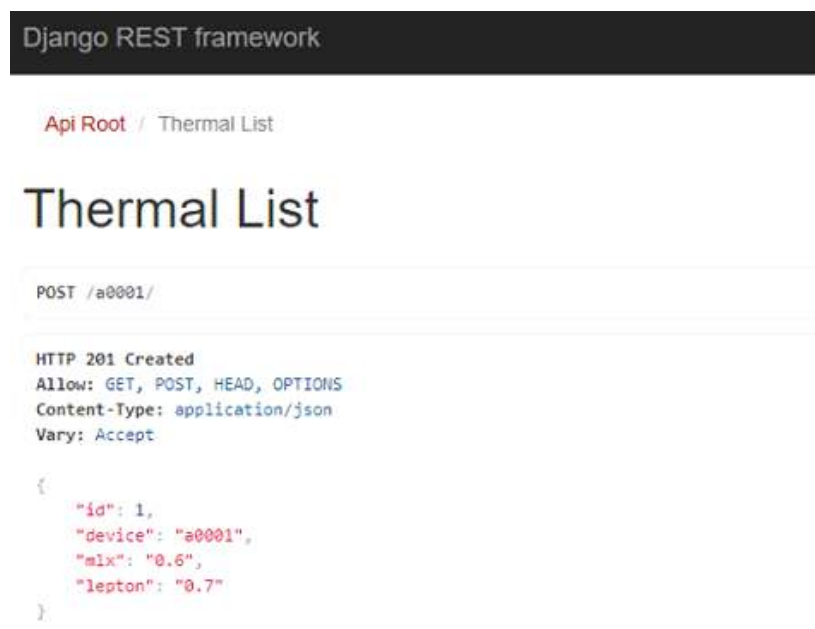
데이터 분리는 학습 데이터 7, 테스트 데이터 3으로 하였으며 11월 중순부터 12월 초까지 측정한 데이터를 사용하였다. 여기서 문제가 구현이 가을 말부터 겨울 초까지 짧은 기간을 측정하여서 영하 -10도 이하까지 떨어지는 한겨울, 고온 다습한 여름의 데이터가 전무하여 계산이 완벽하지 않다.

## 제 2절 서버 구현

파이썬 기반의 웹 프레임워크인 django와 django REST framework를 사용하여 웹 서버 프로그램을 구현하였다. django는 모델-뷰-컨트롤러(MVC) 패턴을 따르고 있으며 컴포넌트의 재사용성과 플러그인화 가능성, 빠른 개발이 특징이다. 해당 프로젝트에서는 간단하게 json 데이터를 배포하는 것이 목적이므로 빠른 개발을 할 수 있는 django를 사용하였다.

또한 데이터를 채온 측정기기에 단순히 텍스트가 아닌 정의되어 있는 API로 제공하였다. API로 서버 컴퓨터와 채온 측정기기가 상호작용하여 정보를 교환 할 때 원하는 정보를 전달할 수 있다. 여기서 json으로 전송할 수 있는 REST API는 클라이언트와 서버, 리소스로 구성되어 있으며 요청이 HTTP를 통해 관리한다. 이렇게 함으로써 API 사용을 위한 별도의 인프라를 구축할 필요가 없으며 모든 플랫폼에서 사용이 가능하다. 하지만 HTTP Method 형태가 제한적인 것이 단점이지만 해당 프로젝트에서는 Post, Get만을 사용하기 때문에 문제가 되지 않는다.

json을 선택한 이유는 json이 데이터의 송수신을 자바스크립트 객체로서 수행할 수 있게끔 하는 가벼운 문자열 데이터 표현식이다. 비슷한 것으로 xml이 있지만 크기가 비교적 무겁고 json을 사용한 API의 사용이 더 많아서 json을 사용하였다.



[그림 8] Rest API 구현

[그림 8]의 “id”는 목차이다. 보정 온도가 생성될 때마다 “id”가 1씩 늘어나면서 온도 측정기기는 “id”를 통해 최신 데이터를 판단할 수 있다. “device”는 각 기기의 고유 번호이고 “mlx”는 손목 온도를 측정할 열 측정 센서의 보정 온도, “lepton”은 얼굴 온도를 측정할 열 화상 카메라의 보정 온도이다. mlx와 lepton은 각각 사용 부품의 모델명인 mlx90614, Lepton 2.5에서 가져와 사용하였다.

### 제 3절 자동화

타이머 계산에는 Python 작업 예약용 라이브러리 schedule를 사용한다. 해당 라이브러리는 3초, 1분마다 등으로 시간 간격으로 실행을 할 수 있으며 매일 특정 시간, 심지어 특정일까지 지정하여 프로그램을 자동으로 실행시킨다. 다른 라이브러리로는 APScheduler이 있지만 해당 프로젝트에서는 매일 특정 시간에만 실행을 시킬 예정이라 명령어가 직관적이고 사용이 용이한 schedule를 사용한다.

타이머는 아래의 조건에 부합할 때마다 프로그램을 실행한다.

- 매시 3분 (매시 정각 외부 날씨데이터 갱신으로 인한 보정 온도 값 변화)
- 매시 0,5,10,15,20,25,30,35,40,45,50,55분  
(내부 환경 데이터 갱신으로 인한 보정 온도 값 변화)

프로그램이 실행되면 외부 / 내부 환경 데이터가 저장된 데이터베이스에 접근하여 가장 최근의 데이터를 가져온 후 해당 데이터를 머신러닝으로 학습된 모델을 통해 각 보정 온도를 계산한다. 그 후 계산된 보정 온도들은 서버 프로그램의 데이터베이스에 접속하여 데이터를 저장하면 서버가 자동으로 데이터를 갱신한다.

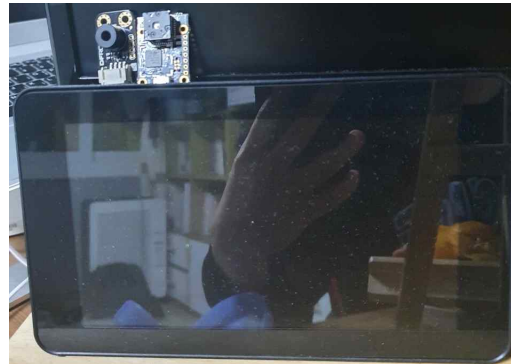
```
<QuerySet [{ 'id': 1, 'device': 'a0001', 'mlx': '8', 'lepton': '7'}, { 'id': 2, 'device': 'a0001', 'mlx': '6.5', 'lepton': '7'}, { 'id': 3, 'device': 'a0001', 'mlx': '6.5', 'lepton': '7'}, { 'id': 4, 'device': 'a0001', 'mlx': '4.5', 'lepton': '2'}, { 'id': 5, 'device': 'a0001', 'mlx': '7.5', 'lepton': '4'}, { 'id': 6, 'device': 'a0001', 'mlx': '7.8', 'lepton': '4'}]>
```

[그림 9] 갱신되는 데이터

## 제 4절 테스트

### 테스트용 온도 측정 기기

- Board : Raspberry PI 4 8GB
- OS : Raspbian
- 열 측정 센서 : MLX90614
- 열 화상 카메라 : flir Dev kit2
- 온습도 센서 : DHT-11
- 디스플레이 : Raspberry PI Display 7
- 소프트웨어 : getThermal



[그림 10] 구현한 비접촉 체온 측정기기

[그림 10]의 상단 왼쪽에 위치한 것이 손목 온도를 측정하기 위한 열 측정 센서, 오른쪽에 위치한 것이 얼굴 온도를 측정하기 위한 열 화상 카메라이다. 정면에 위치한 디스플레이와 후면에 있는 스피커를 통해 결과를 출력하고 추가로 후면에 온습도 센서를 배치하여 내부의 환경을 측정한다. 이 모든 기능들은 후면의 라즈베리파이가 관리한다.

### 접촉 온도 측정기기

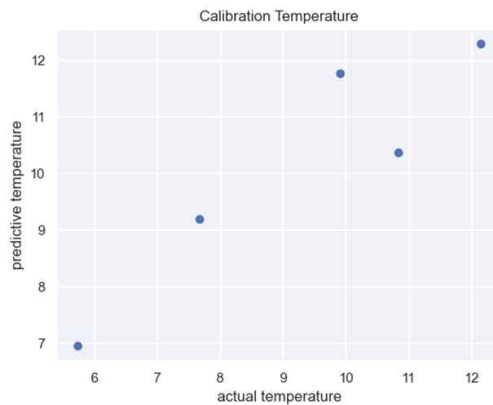
- MT1681

### 테스트

1. 밖에서 10분~20분 대기하고 비접촉 센서로 손목 및 얼굴 체온 측정
2. 그 후 바로 접촉 온도 측정을 한 후 비접촉 측정 온도와 접촉 온도의 차로 보정 온도를 구한 후 데이터베이스에 저장
3. 1,2를 하루에 1~2번 씩 11월 중순부터 12월 초까지 반복
4. 온도를 측정할 때 학습된 데이터를 기반으로 보정 온도를 가져와 합산

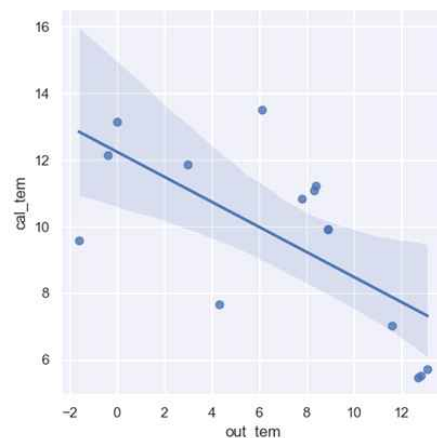
## 제 5절 테스트 결과

현재 보여주는 테스트 결과들은 명확한 결과 및 차이를 보여주기 위해 열 측정 센서인 MLX90614를 사용한 결과만을 보여주고 있다.



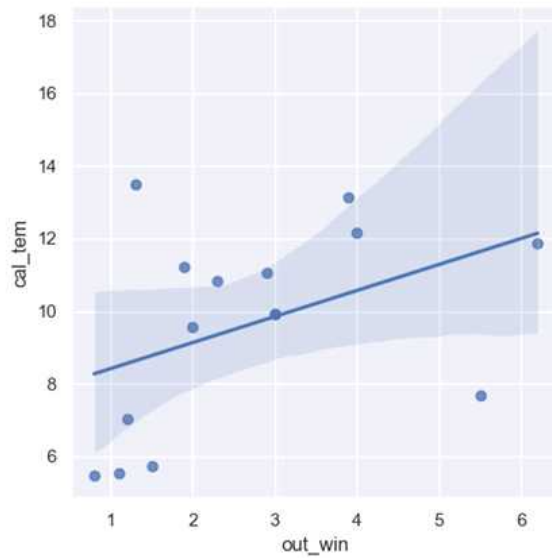
[그림 11] 보정 온도의 정확도

[그림 11]의 x축은 실제 보정 온도이고 y축은 계산된 보정 온도이다. 학습데이터가 8, 테스트 데이터가 2의 비율이라 결과 값이 적다. 하지만 실제 보정 온도와 계산된 보정 온도의 오차 비율은 10~20퍼센트 정도 밖에 되지 않으며 보정 온도를 사용하지 않을 때보다 평균 30퍼센트 정도 향상된 것을 볼 수 있다.



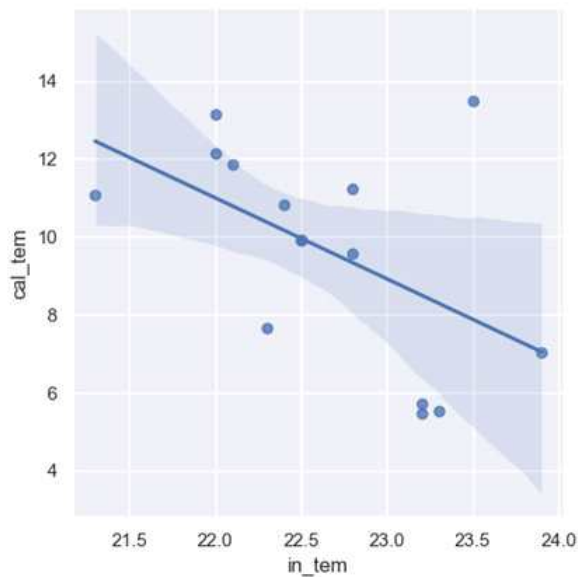
[그림 12] 보정 온도와 야외 온도

[그림 12]의 x축은 야외 온도이고 y축은 보정 온도이다. 야외 온도가 높아질수록 보정 온도는 반비례하여 줄어든다.



[그림 13] 보정 온도와 야외 풍속

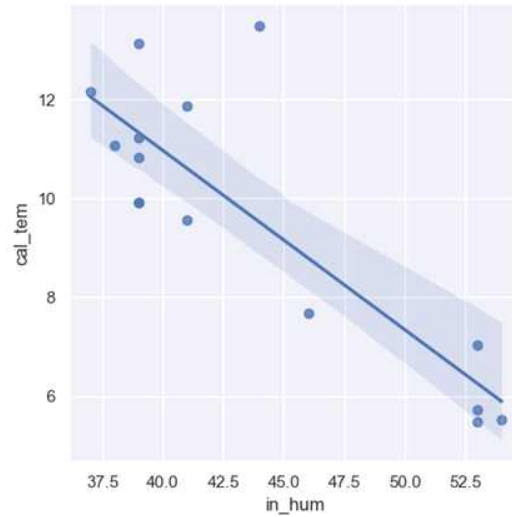
[그림 13]의 x축은 야외 풍속이고 y축은 보정 온도이다. 야외 풍속이 강해질수록 보정 온도는 비례하여 늘어난다.



[그림 14] 보정 온도와 실내 온도

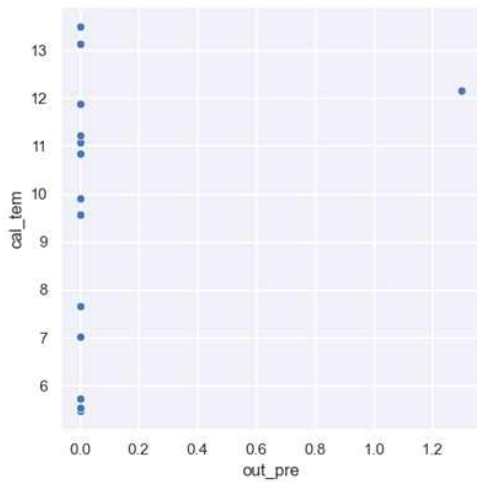
[그림 14]의 x축은 내부 온도이고 y축은 보정 온도이다. 내부 온도가 높아질수록 보정 온도는 반비례하여 줄어든다.



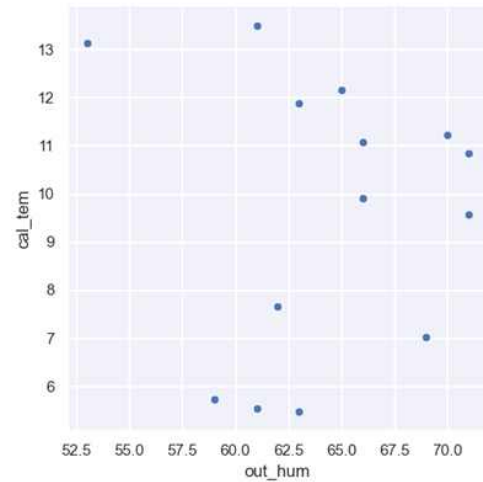


[그림 15] 보정 온도와 실내 습도

[그림 15]의 x축은 내부 습도이고 y축은 보정 온도이다. 내부 습도가 높아질수록 보정 온도는 반비례하여 줄어든다.



[그림 16] 보정 온도와 야외 강수량



[그림 17] 보정 온도와 야외 습도

[그림 16]과 [그림 17]은 보정 온도와와의 연관 관계를 밝혀내지 못하였다. 원인으로 추정하는 것은 부족한 데이터라 판단한다. 강수량 데이터는 측정한 데이터에서 강수량 데이터가 너무 적어 판단이 힘들며 외부 습도 같은 경우 너무 건조하거나 다습한 경우가 없어서 제대로 판단하기 어려운 것이라 생각된다.

## 제 5장 문제점 및 결론

### 제 1절 문제점

측정에 사용된 하드웨어 센서들의 정확도가 낮아 기존에 판매되고 있는 체온 측정기기에 비해 측정 온도가 낮게 나온다. 평균적으로 판매되는 체온 측정 센서는 정상 측정 온도가 34도에서 36도 내외인 반면에 사용된 열 측정 센서 중 가장 정확도가 높은 것이 30~33도로 측정이 된다. 이것은 사용된 센서와 열 화상 카메라가 의료용이 아닌 산업용이라 정확도가 낮게 나온다. 하지만 의료용은 별도의 허가가 필요하기 때문에 따로 구매해서 테스트를 할 수가 없었다.

그리고 다양한 환경에서 실험하지 못한 것이 있다. 현재 사용된 데이터는 11월 중순부터 12월 초까지 짧은 기간 동안 테스트를 하였다. 그래서 보정 온도와 환경 조건의 관계를 제대로 파악하지 못한 것이 많아 해당 논문을 조금 더 일찍 기획을 하여 많은 학습데이터를 모았다면 지금보다 더 좋은 결과가 나왔을 것이다.

### 제 2절 결론

현재 열 화상 카메라와 열 센서들의 측정 성능은 계속 향상되고 있다. 하지만 비 접촉 온도 측정의 구조적인 한계 상 접촉 온도 측정과의 정확도는 근소하게나마 차이가 날 것을 생각해보면 주변 온도를 통한 보정 온도는 이러한 차이를 줄여 접촉 온도 측정과 근접하게 측정 온도를 보여줄 것이라 생각된다.

## 참 고 문 헌

- 기계학습을 통한 얼굴인식 및 체온측정 출입관리 시스템  
(김진하, 김응곤, 한국전자통신학회 논문지 = The Journal of the Korea Institute of Electronic Communication Sciences v.16 no.1 , 2021년, pp.197 - 202 )
- 주변 환경 온도 변화를 고려한 열화상 온도 데이터의 보정 알고리즘 설계  
(송성호, 한국전기전자학회 2021년 6월 pp.261 - 266)
- 산업용 열화상 가이드북  
[https://www.flirmedia.com/MMC/THG/Brochures/T820264/T820264\\_KR.pdf](https://www.flirmedia.com/MMC/THG/Brochures/T820264/T820264_KR.pdf)
- 모두의 데이터 분석 with 파이썬  
(송석리, 길벗)
- 배프의 오지랖 Django로 쉽게 배우는 파이썬 웹프로그래밍  
(배프, 디지털 북스)
- Red Hat  
<https://www.redhat.com/ko>
- 선형 대수와 통계학으로 배우는 머신러닝 with 파이썬  
(장철원, 비제이퍼블릭)