

---

Wi-fi 기계학습을 통한 공간의 혼잡도 분석 및 AR 서비스 개발

---



팀대표	성명	곽선우	학번	151798
	전공/학년	소프트웨어공학/4학년	E-mail	so2009@naver.com
멤버(1)	성명	김솔	학번	160767
	전공/학년	소프트웨어공학/4학년	E-mail	rlathf718@naver.com
멤버(2)	성명	최영수	학번	134247
	전공/학년	소프트웨어공학/4학년	E-mail	wimalnel@naver.com

---

지도교수 : 최덕재 교수님

최덕재

---

제출일 : 2020/05/15

---

Chonnam National University

## 배경 및 문제

Wi-fi가 흔해진 요즘 시대에 대부분의 시설에는 무선랜 인프라가 구축되어 있다. 사람이 많이 모이는 곳이라면 무선랜 데이터는 빅데이터가 될 것이고 이를 처리한다면 의미 있는 정보를 얻을 수 있다. 학내 시설 중 도서관은 학생들의 사용 빈도가 높고 머무르는 시간이 길어 많은 무선랜 데이터가 수집된다. 또한 학습을 위한 공간인 만큼 학습 효율을 높이기 위해 쾌적한 환경은 중요한 요소이다.

현재 전남대학교 도서관을 이용할 때는 발권시스템을 통해 자리 대여 상태를 확인할 수 있지만 각 열람실에 실제 사람이 있는지, 소음은 얼마나 되는지, 공기는 쾌적한지 등 이와 같은 정보를 접할 수 있는 방법이 없기 때문에 학생들은 쾌적한 환경의 열람실을 찾기 위해 직접 확인해보고 자리를 선택한다. 하지만 시간에 따라 변하는 혼잡도와 공기질 때문에 이용 중인 좌석이 나중에 마음에 들지 않는 경우, 다시 내려가 자리를 선택하는 경우가 빈번히 발생한다.

또한 교내에서 실시한 대학종합만족도조사 중 도서관만족도에 따르면 열람실 총 4,900석을 일평균 약 2,800여명이 이용하고 있으나, 시험기간 동안에는 평상시 대비 약 2배 정도의 이용자들이 집중되면서 좌석 부족 현상을 초래하는 것으로 나타났다.

이러한 점에 주목하여 전남대학교 도서관에서 수집되는 한 달 평균 약 500만건 이상의 무선랜 데이터를 사용하고자 한다. 대량의 데이터에 추가로 공간별 AP의 위치, 사용자 인증정보(Radius) 데이터, 기상 관측 데이터, 공기질 데이터를 결합한다면 열람실별 혼잡도 수치라는 의미 있는 값을 생성 할 수 있다. 이러한 빅데이터를 사람이 일일이 분석하여 결과 값을 내기에 한계가 있기 때문에 기계학습을 통해 데이터 속에 존재하는 패턴을 찾아 혼잡도와 공기질의 미래 값을 추출하고자 한다. 이 값을 학생들에게 제공함으로써 도서관 사용에 대한 불편함을 줄이고 학습능률을 높이는 데 도움을 주는 서비스를 만들 수 있다. 더불어 혼잡도 정보를 기존의 시험기간 강의실 대여 서비스와 연계하여 도서관의 혼잡도가 추가적인 좌석확보를 요구하는 일정 기준치를 넘어가면 학교에 관련 정보가 전달되고 단과대학별 강의실 및 학습실을 개방한 후 알림서비스로 공지함으로써 인원 분배를 통해 비용 투입 없이 열람실 확충 효과를 유도하고자 한다.

결과 값을 제공하는 방법으로는 증강현실 기술(AR)을 이용하여 사용자들의 접근성을 높이고 직관적이며 흥미롭게 서비스를 제공한다. 어플을 이용하여 도서관 내에 직접 오지 않아도 지나가던 중에 카메라로 건물을 비추면 AR 콘텐츠를 통해 정보를 얻을 수 있도록 한다.

## 목표

- 공간별 가공한 Wi-fi 데이터, 공기질 데이터, 날씨 데이터를 이용하여 혼잡도 분석 및 예측 알고리즘을 개발한다.
- 혼잡도와 공기질 정보를 나타낼 AR 콘텐츠를 제작한다.
- 추가 학습공간 확보를 위한 강의실 개방 시스템을 구축한다.
- AR 콘텐츠와 개방 시스템을 결합한 어플리케이션을 제작한다.

### 1) 머신러닝을 통한 데이터 분석 및 예측

- 현재와 미래의 시간별 예측 혼잡도를 구하기 위해서 머신러닝을 위한 오픈소스 라이브러리를 이용한다.
- AP의 특정 칼럼 값을 이용하여 개인당 여러 기기를 사용하는 경우, 내부적으로 데이터가 중복된 경우를 제거한다.
- 시간에 따른 AP 접속 수와 공기질 데이터, 날씨 데이터를 시계열 데이터에 따라 분석하고 기계학습으로 1시간 단위로 예측한다.
- 예측된 AP 접속 수를 기준에 따라 혼잡도 5개의 클래스로 분류한다.
- 머신러닝한 결과를 JSON 파일로 저장하고 Firebase로 전송한다.

### 2) AR 어플리케이션

- 증강현실을 구현하기 위해서 Unity의 Vuforia SDK를 사용한다.
- GPS 위치 정보(위도, 경도, 고도)를 가진 3가지 3D Building Object를 제작한다.
- Object 클릭 시 열람실 정보를 제공할 AR 메뉴를 개발한다.
- 5개 클래스로 분류된 혼잡도를 표현하기 위한 Emoticon Animation 및 다양한 이펙트를 개발한다.



< AR 서비스 예시 >

## 요구사항 목록(일렬번호)

01. 추가적인 장비 설치 없이 기존 장비로 수집되는 Wi-fi 데이터를 활용하여 공간별 혼잡도를 분석 및 예측하도록 한다.
02. Wi-fi Raw 데이터에서 한 사람을 특정하여 식별할 수 없도록 학번 등의 정보는 해시 처리하여 보안성을 유지한다,
03. 한사람이 노트북, 핸드폰, 태블릿 pc등 여러 기기를 사용하여 Wi-fi에 접속 시 발생하는 중복 케이스를 배제할 수 있도록 한다.
04. 사용자가 예측 값을 신뢰할 수 있도록 공간별 혼잡도 예측 정확도가 최소 70% 이상 나오도록 한다.
05. 공간 내 인원이 아닌 외부를 지나가던 사람이 Wi-fi에 접속되는 경우를 배제할 수 있도록 한다.
06. 열람실별 재실인원에 따른 혼잡도를 바탕으로 건물의 전체적인 혼잡 상태도 알 수 있도록 한다.
07. 혼잡도 외에도 공기질 등 부가적인 다른 정보들도 제공하도록 한다.
08. 전남대학교 도서관, 정보전산원, 공대7호관 3가지 건물들을 3D Object 형태로 GPS 위치정보에 맞게 나타나도록 한다.
09. 사용자들이 조작하기 간편하고 재미를 느낄 수 있도록 GUI를 구성한다.
10. 도서관이 매우 혼잡할 시 그 상황을 바로 알 수 있도록 알림기능을 구현하도록 한다.

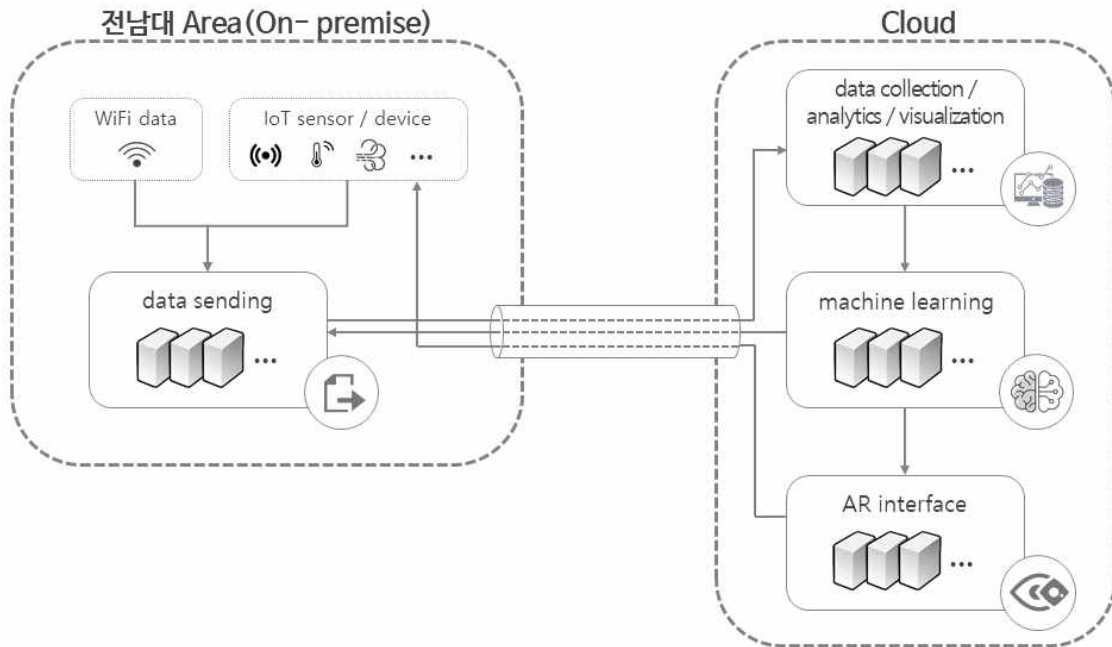
## 일정 (간트차트)

[illegible]

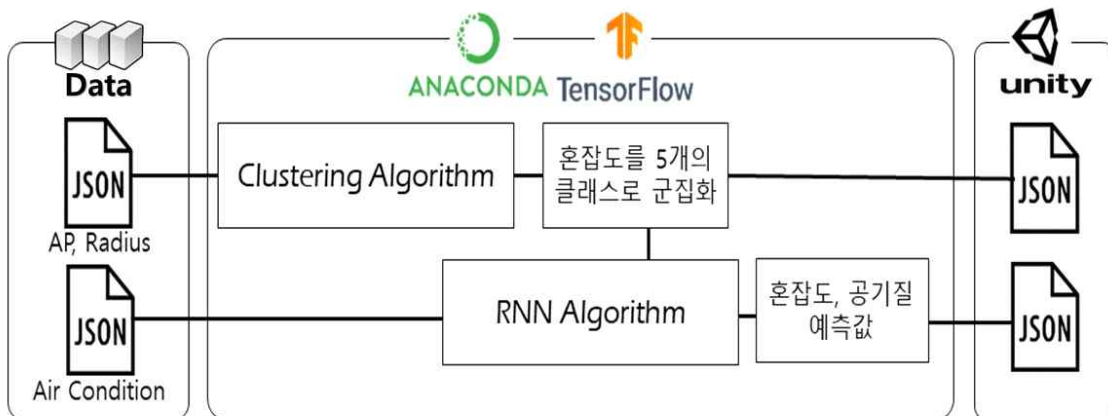
세부 과제 수행내용	추진일정(2020년)									
	4월				5월				6월	
	2주	3주	4주	5주	1주	2주	3주	4주	1주	2주
간트 차트 개선 및 개선점 논의(담당자 : 곽선우)										
- 간트 차트 작성										
- 기존의 결과물에서 개선할 점 논의										
혼잡도 알고리즘 개선(담당자 : 김솔)										
- 추가적인 데이터를 삽입으로 알고리즘 성능 개선										
- 시험 및 검증										
AR 콘텐츠 개선(담당자 : 최영수)										
- 어플 실행시 나타나는 이펙트 개선 및 보완										
- 시험 및 검증										
중간 보고 준비(담당자 : 곽선우)										
- 진행상황 확인 및 보고서 작성										
- 보고서 피드백 반영										
발표 영상 제작(담당자 : 최영수)										
- 프리미어 활용 영상 제작 공부										
- 구현 영상 제작										
최종 발표 준비(담당자 : 김솔)										
- 최종 영상 확인 및 보고서 작성										
- 발표 준비										

## 설계 (아키텍처)

### - 전체 구성도



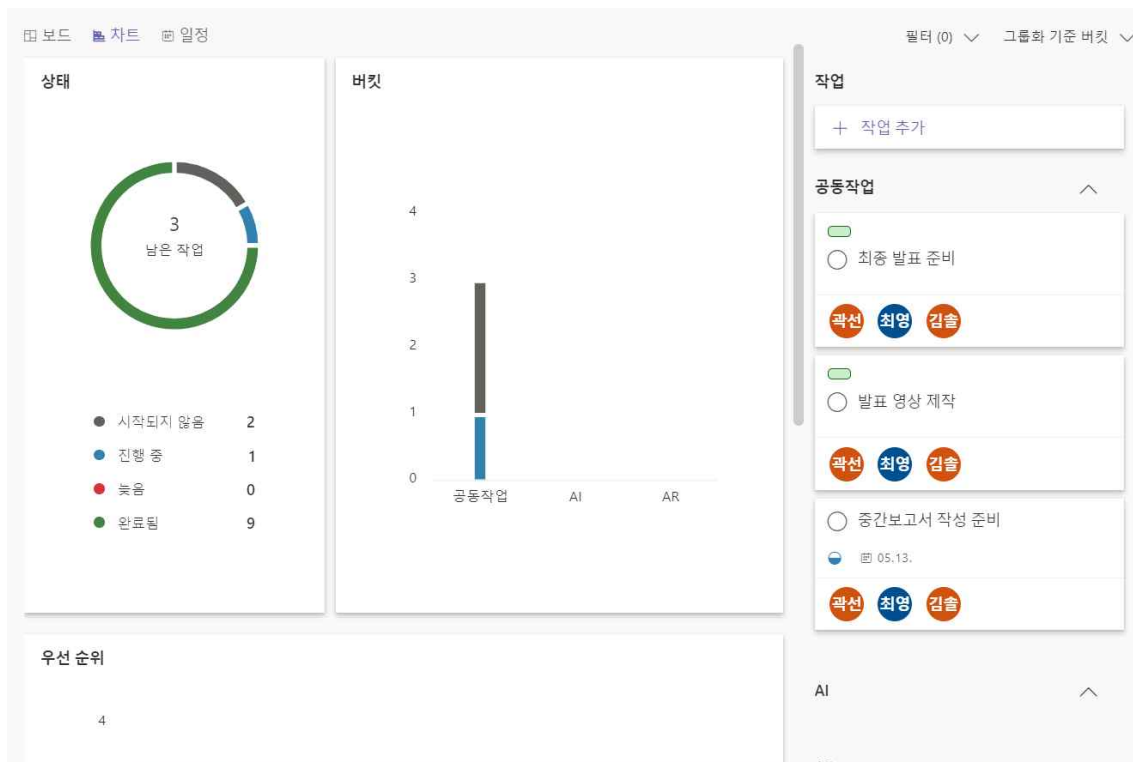
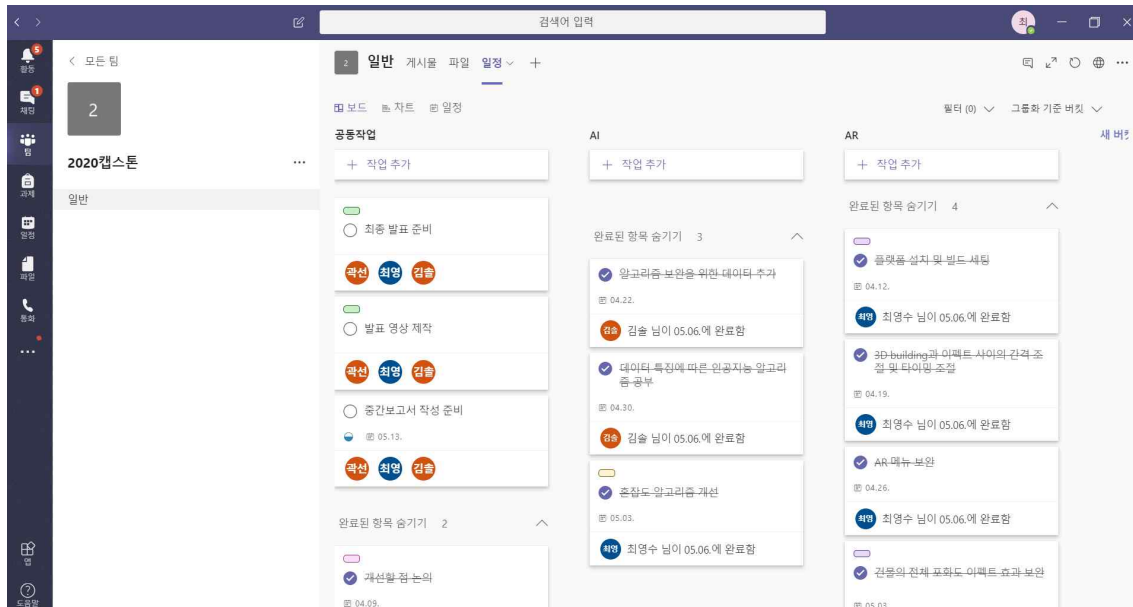
### - 머신러닝 구성도



## 진행상황

### ● 협업과정

Microsoft Teams 사용



- 세부 일정관리 및 진행을 위해 Microsoft Teams를 사용했다.
- 일정과 기간을 추가하였고 팀원의 역할에 따라 배분했다.



## ● 혼잡도 알고리즘 개발

```

for idx1, row1 in date_data.iterrows():
    queue1.queue = [] #username
    queue2.queue = [] #acctstarttime
    queue3.queue = [] #acctsessiontime
    queue4.queue = [] #place
    queue5.queue = [] #idx

    for idx, row in ap_data.iterrows():
        timestamp = row.acctstarttime
        if row1.date, hour+1 < row.acctstarttime, hour:
            break

        while(date_data.iat[idx1,0] <= timestamp) and (timestamp < date_data.iat[idx1+1,0]):
            if row.username == np.nan:
                break

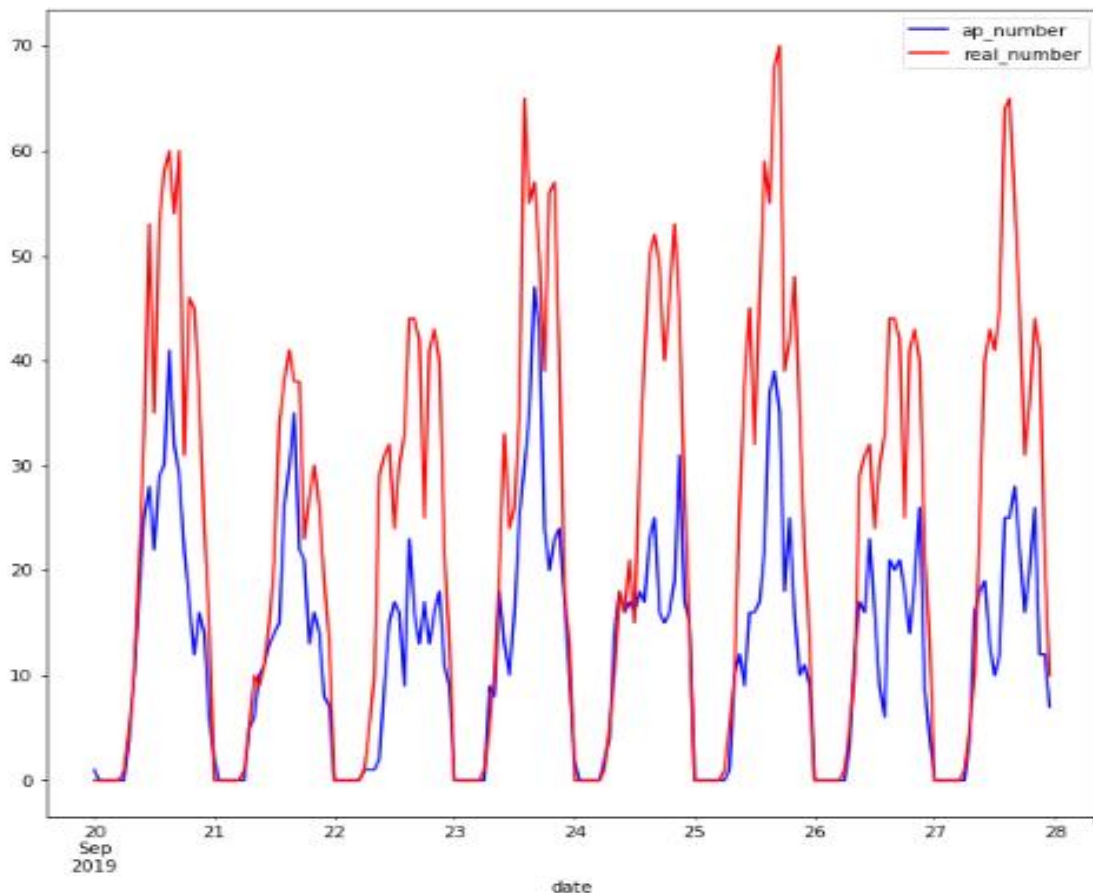
            queue1.push(row.username)
            queue2.push(row.acctstarttime)
            queue3.push(row.acctsessiontime)
            queue4.push(row.place)
            queue5.push(idx)

            if queue1.size() > 1: #큐 안에 1개만 있을때를 위해 2부터 들어가게
                listt = queue1.queue
                if listt.count(ap_data.iat[idx,0]) == 1:
                    break
                else:
                    i = 0
                    for data in listt:
                        if (ap_data.iat[idx,0] == data) and (queue5.queue[i] == idx):
                            i = i + 1
                            continue
                        elif (ap_data.iat[idx,0] == data) and (queue5.queue[i] != idx):
                            if (ap_data.iat[idx,4] < queue3.queue[i]) and (ap_data.iat[idx,0] == queue1.queue[i]):
                                ap_data.iat[idx,0] = np.nan
                                continue
                            elif ap_data.iat[idx,4] >= queue3.queue[i] and ap_data.iat[idx,0] == queue1.queue[i]:
                                ap_data.iat[queue5.queue[i],0] = np.nan
                                i = i + 1
                    break

```

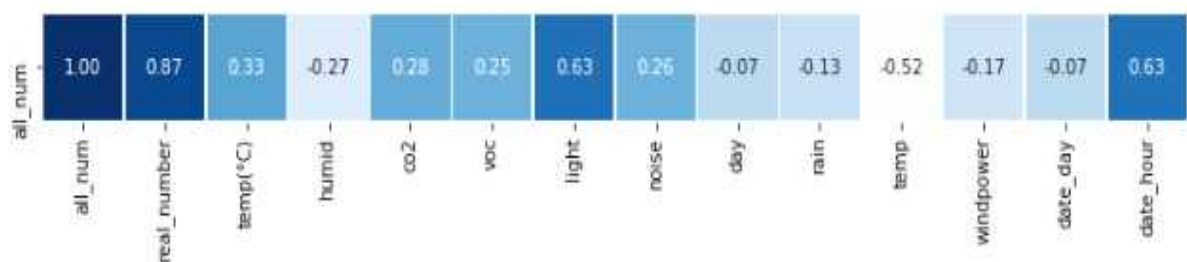
### < 중복 제거 알고리즘 >

열람실마다 AP의 IP 주소를 기준으로 접속자를 나누고 개인당 여러 기기를 사용하는 경우, 내부적으로 데이터가 중복된 경우를 배제하기 위한 알고리즘을 만들었다. 큐를 사용하여 데이터들을 큐에 넣고 사용자가 겹치면 시간 데이터를 비교하여 최초 접속 시각이 비슷하면 접속시간이 더 짧은 데이터를 삭제했다.



< AP 접속자 수, 재실 인원수 비교 >

위 그래프는 일주일 동안의 열람실별 AP 접속자 수와 실제 재실 인원수의 그래프이다. 값의 차이가 보이지만, 시간대별 사람 수가 증가할 때 AP 접속 수도 증가하고 사람 수가 낮아질 때 AP 접속자 수도 낮아지는 흐름을 볼 수 있었다. 이 결과를 바탕으로 AP 접속자 수와 실제 재실 인원수는 높은 상관관계가 있을 것으로 추정하였다.



< 데이터 간 상관관계 >

데이터 분석을 위해서 열람실별 ap 접속자 수, 재실 인원수, 공기질 데이터와 실제 외부 날씨 데이터를 시각화 모듈을 사용하여 상관관계를 확인하였다. 위에서 추정한 것과 같이 ap 접속자 수(all\_num)와 실제 재실 인원수(real\_number)는 0.87이라는 높은 상관관계를 보였다. ap 접속자 수를 기준으로 재실 인원수, 시간, light, 외부 기온, 내부 기온 순으로 상관관계가 높았다.

```

from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import VotingRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import svm

n_train = 24*17
y_t = []
y_p = []

def eval_on_features(features, target, regressor):
    X_train, X_test = features[:n_train], features[n_train:]
    y_train, y_test = target[:n_train], target[n_train:]
    y_t.append(y_test)
    regressor.fit(X_train, y_train)
    print("훈련 세트 R^2: {:.2f}".format(regressor.score(X_train, y_train)))
    print("테스트 세트 R^2: {:.2f}".format(regressor.score(X_test, y_test)))
    y_pred = minmax_scale(regressor.predict(X_test))
    y_pred_train = minmax_scale(regressor.predict(X_train))
    y_p.append(y_pred)
    plt.figure(figsize = (15,15))

    plt.xticks(range(0, len(X), 8), xticks_name, rotation = 300, ha="left")

    plt.plot(range(n_train), y_train, label = "훈련")
    plt.plot(range(n_train, len(y_test) + n_train), y_test, '-', label = "테스트")
    plt.plot(range(n_train), y_pred_train, '--', label = "훈련 예측")

    plt.plot(range(n_train, len(y_test) + n_train), y_pred, '-', label = "테스트 예측")
    plt.legend(loc = (1.01, 0))
    plt.xlabel("날짜")
    plt.ylabel("대여횟수")

X_hour_week = np.hstack([df.hour.values.reshape(-1,1), df.기온.values.reshape(-1,1),
                          df.temp.values.reshape(-1,1), df.light.values.reshape(-1,1),
                          df.voc.values.reshape(-1,1), df.all_num.values.reshape(-1,1)])

reg1 = GradientBoostingRegressor(random_state=10, n_estimators=15, max_depth = 4,
                                 learning_rate= 0.1, min_samples_leaf= 9, max_features= 1.0)

reg2 = RandomForestRegressor(random_state=10, n_estimators=10)

ereg = VotingRegressor(estimators=[('gb', reg1), ('rf', reg2)])

eval_on_features(X_hour_week, y , ereg)

```

---

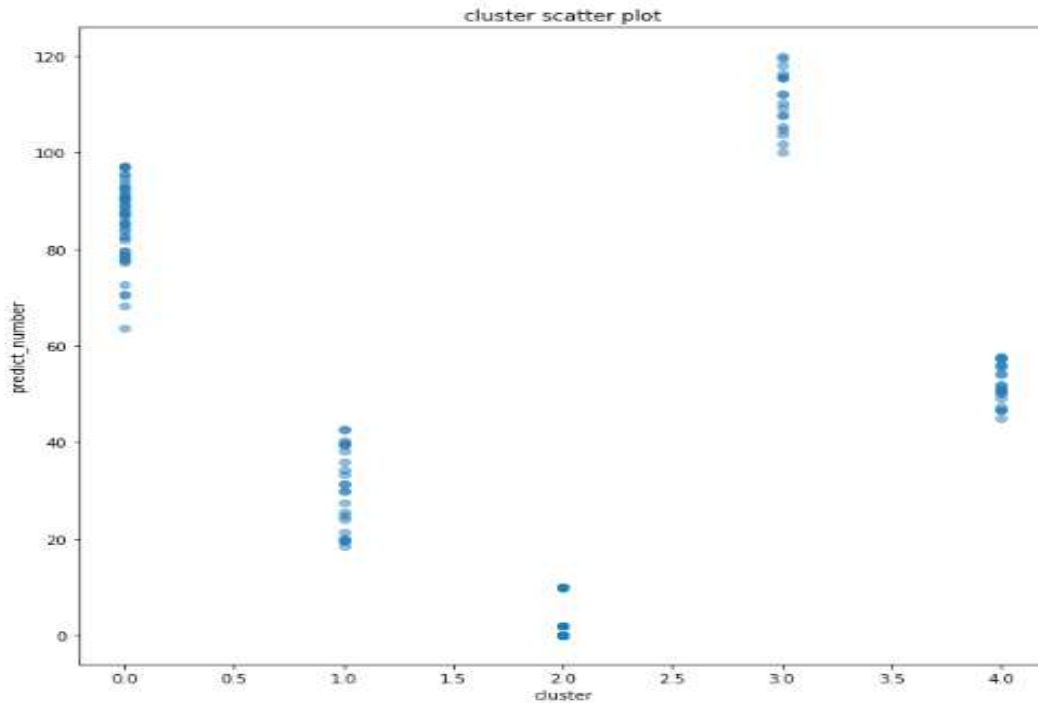
```

훈련 세트 R^2: 0.95
테스트 세트 R^2: 0.85

```

### < 예측모델 알고리즘 >

학습할 때 연속적인 숫자(인원수)를 예측하기 위해 회귀 알고리즘인 Gradient-BoostingRegressor를 사용하여 1시간 단위의 예측 재실 인원수를 얻었다. 입력 값으로 상관관계가 높은 공기질 데이터, ap 접속자 수, 시간 데이터를 사용하였고 출력 값으로 실제 재실 인원수를 사용하여 모델링을 하였다. 그 결과, regression model의 성능을 나타내는 값인 결정계수( $R^2$ ) 값이 0.85가 나왔다. 결정계수 값이 1로 갈수록 좋은 모델이라고 할 수 있다. 생성한 예측 모델을 통해 1시간 단위로 재실 인원수를 예측하였다.



< 혼잡도 분류 >

```
predict.loc[predict.predict_number<15, 'predict_congest'] = '1'
predict.loc[(predict.predict_number>=15) & (predict.predict_number<40), 'predict_congest'] = '2'
predict.loc[(predict.predict_number>=40) & (predict.predict_number<60), 'predict_congest'] = '3'
predict.loc[(predict.predict_number>=60) & (predict.predict_number<100), 'predict_congest'] = '4'
predict.loc[predict.predict_number>=100, 'predict_congest'] = '5'
```

< 혼잡도 분류 알고리즘 >

예측된 재실 인원수를 혼잡도 5단계로 나누기 위해서 군집화 알고리즘을 사용하였다. 이 그래프를 바탕으로 재실 인원수가 0 ~ 15이면 혼잡도 1, 15 ~ 40이면 혼잡도 2, 40 ~ 60이면 혼잡도 3, 60 ~ 100이면 혼잡도 4, 100 이상이면 혼잡도 5로 기준을 정하였다. 실제 재실 인원수와 예측된 재실 인원수를 특정 기준으로 혼잡도를 5단계로 분류한 후 비교해 본 결과 일주일의 테스트 데이터 168개 중 131개가 같았고 약 78%의 정확도가 나왔다.

## ● AR 어플리케이션

### 열람실 혼잡도 분류

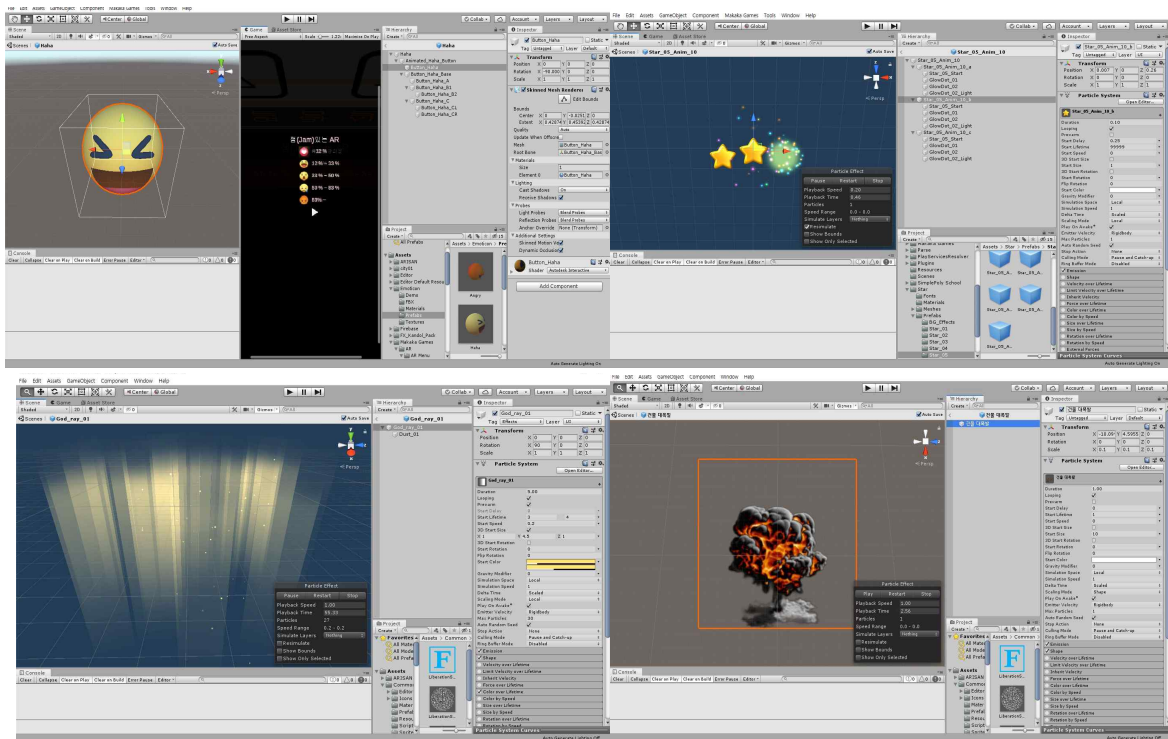
	<b>5단계</b>	약 100명 ~ (열람실 좌석수 120석 기준) 83% ~
	<b>4단계</b>	약 60명 ~ 99명 (열람실 좌석수 120석 기준) 50% ~ 83%
	<b>3단계</b>	약 40명 ~ 59명 (열람실 좌석수 120석 기준) 33.3% ~ 50%
	<b>2단계</b>	약 15명 ~ 39명 (열람실 좌석수 120석 기준) 12.5% ~ 33.3%
	<b>1단계</b>	~ 약 14명 (열람실 좌석수 120석 기준) ~ 12.5%

### 건물 상태도 분류



<b>Jam</b>	도서관이 매우 혼잡하여 추가적인 공간확보 요구 상태
<b>One Star</b>	도서관이 혼잡한 상태 (건물 전체 포화도 80% ~ 95%)
<b>Two Star</b>	도서관의 일반적인 상태 (건물 전체 포화도 40% ~ 80%)
<b>Three Star</b>	도서관이 쾌적한 상태 (건물 전체 포화도 ~ 40%)
<b>Open</b>	추가적으로 개방된 공간에 나타나는 효과

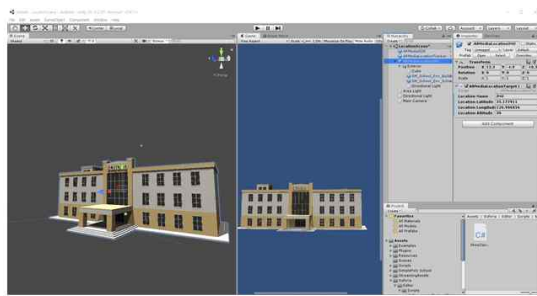
< AR 콘텐츠로 혼잡도 분류 표현 >



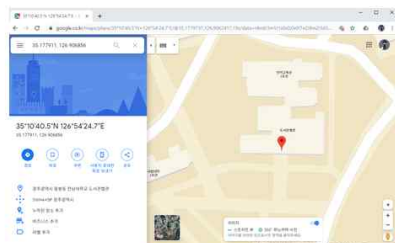
열람실별 재실인원에 따른 혼잡도를 바탕으로 건물의 전체적인 상태를 나타내기 위해 건물 전체 포화도를 기준으로 4단계로 분류했다. 포화도가 0 ~ 40%이면 별 3개, 40% ~ 80%이면 별 2개, 80% ~ 95%이면 별 1개, 95% ~ 100%면 Jam 단계로 기준을 정했다.



재실인원에 따른 5단계의 혼잡도를 나타내는 하트, 웃음, 놀람, 울음, 화냄의 5가지 3D Emoticon Animation과 건물의 전체적인 혼잡도를 표현하기 위해 Unity의 Particle System을 이용하여 건물위에 별 모양 이펙트가 나타나게 하는 Prefab들을 만들었다. 건물이 추가적으로 개방된 상황을 나타내기 위한 별빛이 내리는 이펙트, 건물이 매우 혼잡하여 추가적인 공간확보를 요구하는 상황에서 나타나는 폭발 이펙트를 개발하였다.



Unity 개발환경

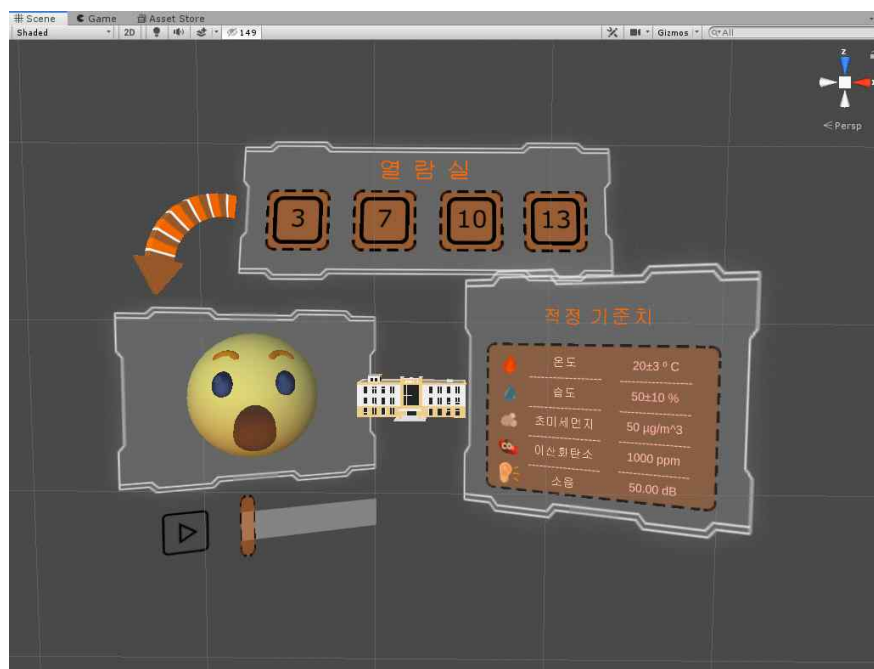


전남대학교 도서관 위치정보

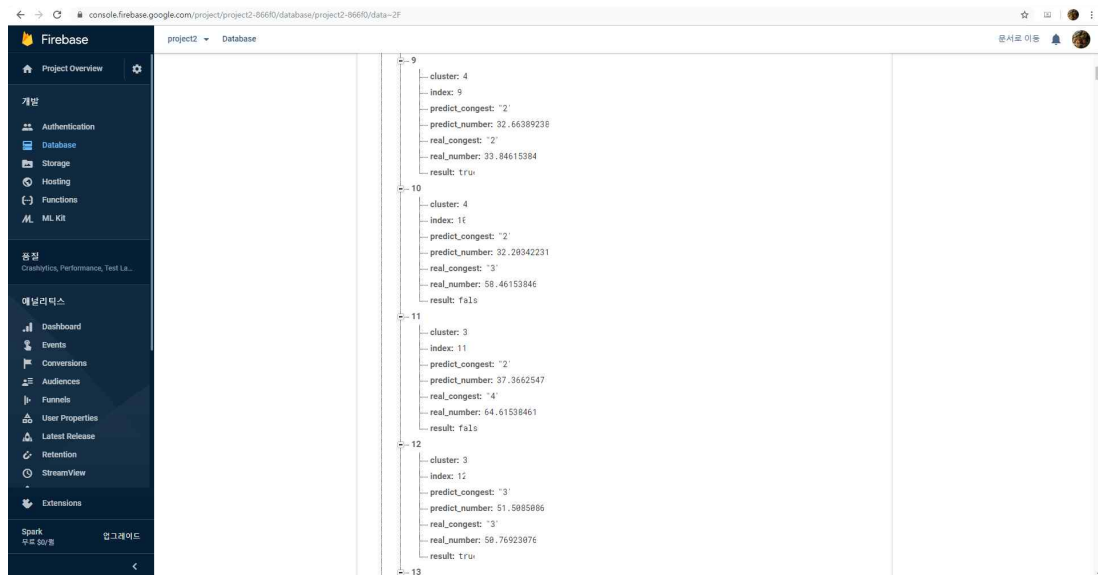


### < 3D Building Object 생성 및 GPS 위치 정보 삽입 >

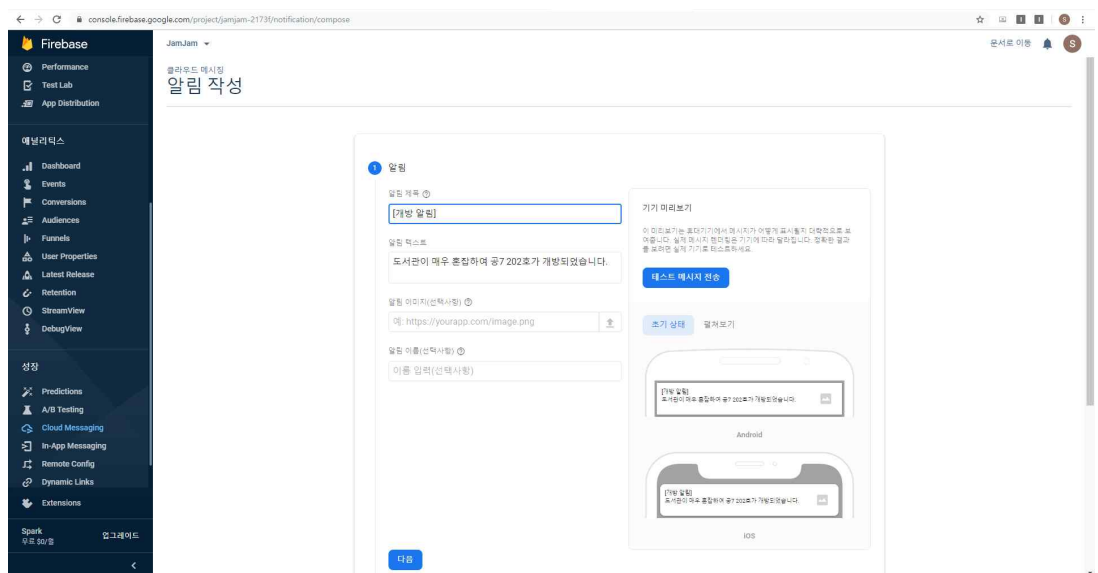
전남대학교 도서관, 정보전산원, 공대7호관 3가지 건물들을 3D Object로 제작하고 Google Map API를 이용하여 GPS 위치 정보(위도, 경도, 고도)를 삽입했다.



오브젝트 클릭시 나타나게 될 전체 콘텐츠를 담은 메뉴를 GUI 형태로 제작했다. 환경부가 정한 도서관 등의 실내오염물질 유지기준 및 권고기준(2018.12월 기준)에 따른 초미세먼지, 이산화탄소의 적정기준치( $50\mu\text{g}/\text{m}^3$ , 1000ppm)를 제시하고 소음·진동 관리법에 따른 도서관 소음규제 기준치인 50dB를, 겨울철 실내 온·습도의 표준범위(온도  $20\pm 3^\circ\text{C}$ , 습도  $50\pm 10\%$ )을 제공하여 사용자가 데이터를 판단하는데 도움이 되도록 했다.



머신러닝을 통해 분류한 혼잡도 값과 공기질 데이터 값은 Json파일로 Firebase에 구축한 Database에 저장된다. Firebase와 Unity와 연동하여 혼잡도와 공기질을 불러올 수 있게 했다.



Firebase의 cloud Messaging 시스템을 이용하여 도서관이 매우 혼잡해 다른 단과대학의 강의실을 개방한 경우 사용자에게 알람이 가도록 했다.