[인공지능프로젝트] Team5 최종 보고서

1st Author

송강규  
Sungkyunkwan Univ.  
2020312668

sgk1004s@naver.com

2nd Author

최장섭  
Sungkyunkwan Univ.  
2019310036

kgh010529@gmail.com3rd Author

이장엽  
Sungkyunkwan Univ.  
1st line of address

3rd E-mail

4rd Author

한성욱  
Sungkyunkwan Univ.  
2020312275

max882816@gmail.com

**ABSTRACT**

디도(삼성학술정보관)을 이용할 때 SKKU 학술정보관 어플₁을 통해서 자리를 확인할 수 있지만, 발권하지 않고 사용하는 이용자들의 수가 많기에 해당 어플을 통해 좌석 현황을 확인하는 데에는 한계가 있습니다. 실제 좌석 현황을 확인하기 위해 본 팀 프로젝트에서는 간 상관관계를 공간적, 시간적으로 분석하여 예측하는 GCN + GRU 모델을 설계하였다. 해당 팀 프로젝트를 통해 실제 좌석 데이터에 근접하는 예측을 달성하였고, 좌석 간 상관관계를 GNN으로 접근하고자 하는 시도를 하였다는 데 의의가 있다.

**CCS Concepts**

• Computing methodologies → Machine learning;

• Applied computing→ Law, social and behavioral sciences;

**Keywords**

Machine learning; Graph Neural Network; Graph Convolutional Network; Gated Recurrent Unit; Sequential prediction

# INTRODUCTION

디도(삼성학술정보관)를 시험기간에 이용하다 보면 원하는 좌석의 자리가 없는 경우가 많이 있습니다. 현재 SKKU 학술정보관 어플을 통해서 자리를 미리 확인할 수 있지만, 발권하지 않고 사용하는 이용자들의 수가 많기에 해당 어플을 통해 좌석 현황을 확인하는 데에는 한계가 있습니다.

해당 프로젝트를 통해서 날짜 정보와 학술정보관 자리 정보를 입력으로 받아 좌석 현황을 예측하게 됩니다. 따라서 학생들은 이용자가 발권을 하지 않아 좌석 정보가 불분명한 상황에서도 현재 좌석 현황을 대략적으로 파악할 수 있고, 이를 통해 원활한 도서관 이용을 할 수 있을 것입니다.

또한, 도서관 내에서 시간대에 따라 여유있는 좌석 현황 정보를 제공받음으로써 이용 시기와 목적이 다른 학생들에게 맞춤형 정보를 제공할 수 있을 것입니다.

본 보고서에서는 실제 좌석 현황을 예측하기 위한 접근으로 regression과 GNN 두 가지 방법을 소개합니다. 특히 GNN에서는 GCN과 GRU를 결합한 방식으로 공간적, 시간적 정보를 최대로 활용하는 것을 새롭게 제안합니다.

# Related Work

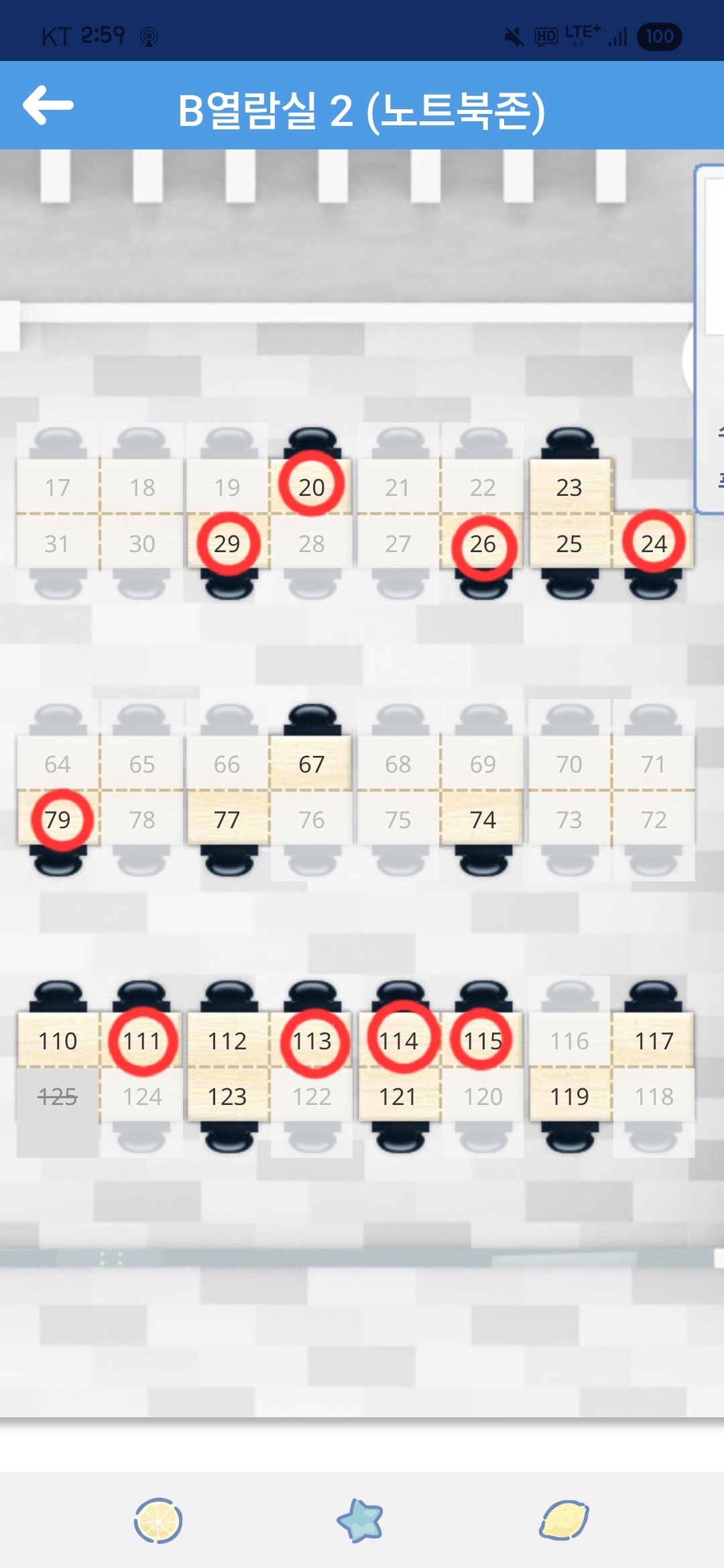


Figure 1. The SKKU Academic Information Center app UI and the seat currently in use without being checked in (red circle)

.

기존 SKKU 학술정보관 어플에서는 Figure 1의 좌측 화면과 같이 좌석들의 배치도와 각 좌석의 발권 상태를 표시한다. 좌석의 색이 옅은 노란색이라면 발권이 가능한 상태이고, 좌석의 색이 하얀색이라면 이미 발권이 된 상태이다.

이렇듯 기존에 존재하는 어플에서도 어떤 좌석이 발권이 된 상태인지는 알 수 있다. 하지만, 실제로는 발권을 하지 않고 좌석을 이용하는 이용자들이 많기 때문에, 실제 좌석 이용 현황과 발권 형황에 큰 차이가 존재하게 된다.

Figure 1의 오른쪽 화면에서 그 차이를 확인할 수 있다. 빨간 동그라미는 미발권 좌석이지만 이용자가 위치해있는 좌석들을 표시한 것이다.

SKKU 학술정보관 어플만 보고 도서관 좌석을 이용하기 위해 방문한 사용자는 어플의 발권 현황과는 다른 좌석 이용 현황으로 인해 도서관 이용에 차질을 겪을 것이다.

본 팀 프로젝트에서는 이러한 발권 현황과 실제 이용 좌석 현황간의 gap을 Machine Learning을 통한 모델 학습을 통한 예측으로 극복하는 것을 목표로 한다.

# Methodology

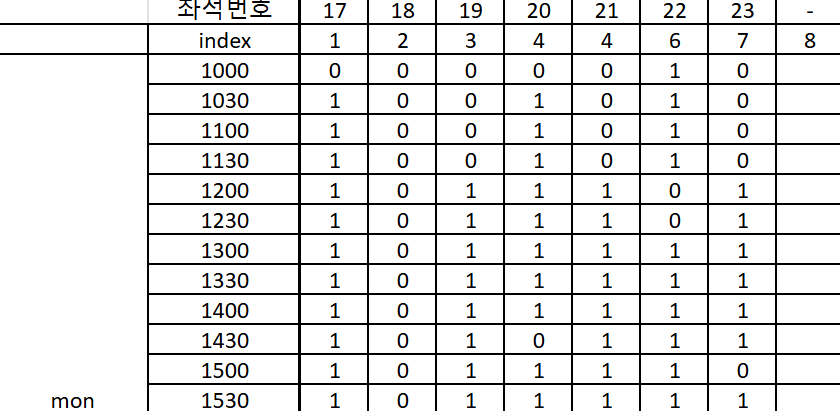
## Data Collection

|  |  |  |
| --- | --- | --- |
|  | **Check-in** | **Non-Check-in** |
| In use | (a) | (b) |
| Not in use | (c) | (d) |

Table 1. Four types of seat statuses

Figure 2. Example of collected data

.



기존 SKKU 학술정보관 어플로 파악할 수 있는 정보는 Table 1에서 (a)+(c)와 (b)+(d)에 해당되는 정보이고, 해당 프로젝트에서 예측 결과값으로 얻고자 하는 정보는 (d)이다. 때문에 데이터 수집 시에는 어플 화면을 캡처하여 (b) 혹은 (d)를 기록하는 형식으로 진행하였다. ((a)+(c)와 (b) 혹은 (d)가 주어진다면 (d)를 알 수 있기 때문에)

또한, 각각의 자리 이용 유무가 아닌 인원수를 통틀어서 기록하여서는 추후에 있을 GNN Model 데이터에 적합하지 못하다고 판단되어, Figure 2에서와 같이 모든 자리에 대해서 이용 상태를 시간대별로 기록하였다. (1은 사용 불가 좌석, 즉 (a)혹은 (b)혹은 (c)에 해당되는 좌석을 의미하고, 0은 사용 가능 좌석, 즉 (d)에 해당되는 좌석을 의미한다.)

모든 데이터 수집은 삼성학술정보관 지하 1층 B 열람실 2에서 실시했다.

하루에 10시~13시, 13시~16시 오전 오후 데이터를 30분 간격으로 수집하여 하루 당 12개 좌석 데이터를 수집하였다. 5월 26일부터 6월 6일까지 총 12일동안 데이터 수집을 진행하였고, 총 280개의 데이터를 수집하였다.

### Data Collection Protocol

1. 학술정보관 어플의 좌석 현황 스크린 샷을 찍는다.
2. 열람실을 돌아다니며 미발권 좌석에 앉아있는 인원 수를 체크한다.
3. 실제 이용 중인 좌석 수 = 발권 좌석 수 + 미발권 인원 수

## Data Augmentation

[추후 작성]

## Regression Model

데이터 수집 정보와 수집 시에 시간 정보를 활용하여 regression 모델을 구현하였다. 날짜 데이터를 시험기간 여부, 요일 feature로 변환하였다. 시험기간 여부는 임의로 5월 28일을 기점으로 그 전은 0 (시험기간 아님), 그 후는 1 (시험기간)로 설정하였다. 요일 feature는 월요일부터 금요일까지 순차적으로 1부터 5까지의 값을 부여하였다.

모델의 입력으로 시험기간 여부, 요일 feature와 어플 상 발권 현황 좌석 수(Table 1에서 좌석 상태가 (a)혹은 (c)인 좌석의 수)를 사용하고, 출력으로 실제 이용 좌석 수를 예측하였다.

따라서, 현재 날짜와 시간 데이터와 SKKU 학술정보관 어플 상의 발권 현황을 이용하여 실제 해당 공간의 혼잡도를 예측하도록 하였다.

### Model Structure

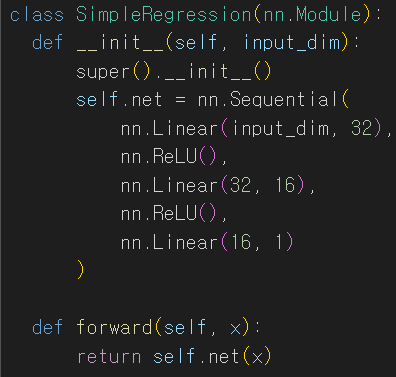


Figure 3. Regression model structure

.

Figure 3와 같은 모델 구조를 채택하였다. Pytorch로 작성한 Multi Perceptron 기반의 regression model로, input\_dim만큼의 차원을 입력으로 받고, 1차원의 출력을 반환한다. 총 세 개의 Linear Layer를 가지고, 각 Layer에서 32차원 16차원 1차원으로 순전파가 진행된다.

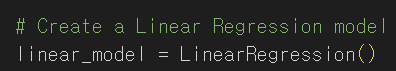


Figure 4. The structure of the linear egression model for comparison

.

우리가 고안한 모델(Figure 3)의 성능을 확인하기 위한 비교군으로 sklearn 라이브러리의 LinearRegression 모델을 사용하였다. (Figure 4 참고)

### Loss Function and Hyperparameters

손실함수로는 MSE loss를 사용하였고, learning rate는 0.001, epoch는 10000으로 설정하였다.

optimizer로는 Adam을 사용하였다.

### Limitation

해당 regression model은 각 좌석의 상태를 고려하지 않고 총 좌석 수를 다루기 때문에 사용자의 니즈를 충분히 충족시키지 못한다. 일반적으로 좌석을 선택할 때 밀집한 좌석보다 sparse한 좌석을 선호하는 경향이 있는데, 해당 regression 모델에서는 이러한 경향성을 반영하지 못한다는 한계가 있다.

## GNN Model

[장섭님 부탁드립니다..]

# Experiments

## Environment

실험은 Google Colab에서 진행하였고, T4 GPU를 사용하였다.

## Dataset

앞서 설명한 바와 같이 3주동안 수집한 데이터 약 280여개의 좌석 정보를 데이터로 이용하였다.

[Data augmentation 관련 내용 추가]

다만, regression model의 경우 2주차에 그 한계를 발견하여 GNN model 구현으로 노선을 바꿨으므로 2주차까지의 dataset만 사용하였다.

## Baseline

### Regression Model

기존에 구현되어 있는 관련 모델이 전무하여 Figure 4에서와 같이 구현된 sklearn의 linear regression model을 baseline으로 선정하였다.

### GNN Model

[장섭님 부탁드립니다..]

# Results

## Regression Model

우리가 구현한 모델(Figure 3)의 경우, 초기 100에포크까지 9.02의 training Loss를 기록했고 10000번째 에포크에서 3.49의 training loss를 기록했다. 최종적으로 3.94의 validation loss를 기록했다.

Figure 4의 sklearn의 linear regression model의 경우 4.06의 validation loss를 기록했다.

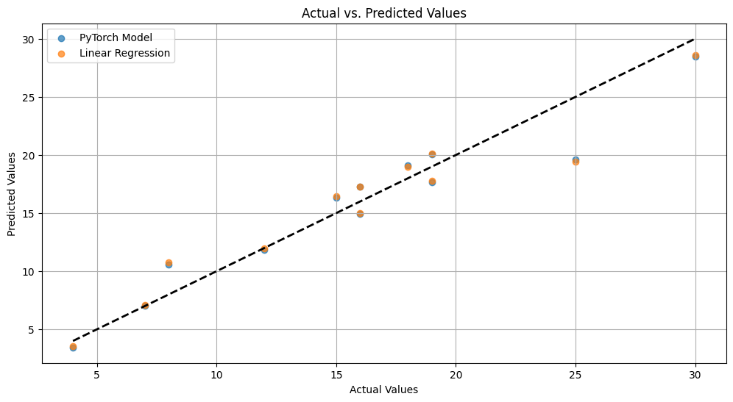


Figure 5. Actual values vs. Predicted Values

.

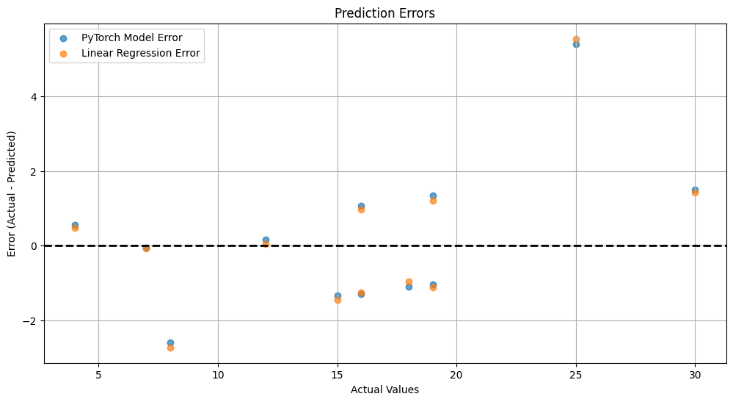


Figure 6. Prediction Errors

.

Figure 5와 Figure 6는 각각 실제 입력값을 바탕으로 모델이 예측한 결과값과 그로부터 계산한 error를 나타낸 그래프이다.

파란 점은 Figure 3의 우리가 구현한 pytorch 모델이고, 주황색 점은 Baseline 모델로 선정한 Figure 4의 sklearn에서 구현된 linear regression model을 나타낸다.

그림에서 볼 수 있듯이 둘의 차이가 미미한 것을 확인할 수 있다.

또한, 정확성 지표로 R-squared를 사용한 결과 우리가 구현한 pytorch 모델은 0.92의 정확성을 보여줬고, sklearn의 Linear regression model은 0.91의 정확성을 보여주었다.

기대에 미치지 못하는 정확도를 보여주었는데 특히 개별 데이터를 살펴 보았을 때 오차가 최대 7까지 튀는 모습을 보여주었다. 이는 아마 input feature가 충분히 시간적 특성을 반영하지 못했기 때문으로 보인다. 날짜데이터와 시간데이터 외에 추가적인 input feature를 발견하지 못했다. 따라서 모델의 정교성이 떨어진 것으로 보인다.

또한, regression model의 구조가 단순한 것 또한 빈약한 성능의 원인으로 보인다. 이전에 model 구조를 복잡하게 하였지만, 구조가 복잡해질수록 수렴이 느려져 학습 시간이 길어지는 경향이 있었다. 따라서 현재의 구조로 정착하게 되었다.

다만, 이런 trade off를 감안하더라도 해당 모델의 구조가 지나치게 단순하고 input feature를 고려하지 않아서 성능이 기대 이하로 나오는 것으로 보인다.

또한 해당 모델 학습에 사용된 데이터셋이 너무 적어서 오버피팅은 관찰되지 않았다.

## GNN Model

[장섭님 부탁드립니다..]

# Conclusion

해당 보고서에서는

# REFERENCES

1. INEK corp. 2024. SKKU 학술정보관 어플 https://lib.skku.edu/#/service/mobile-app.

* SKKU 학술정보관 어플 단어에 아래 첨자 ₁ 추가.