[인공지능프로젝트 \_SWE3032\_41] Team5 Final Report

1st Author

송강규  
Sungkyunkwan Univ.  
2020312668

sgk1004s@naver.com

2nd Author

최장섭  
Sungkyunkwan Univ.  
2019310036

kgh010529@gmail.com3rd Author

이장엽  
Sungkyunkwan Univ.  
Unknown

Unknown

4rd Author

한성욱  
Sungkyunkwan Univ.  
2020312275

max882816@gmail.com

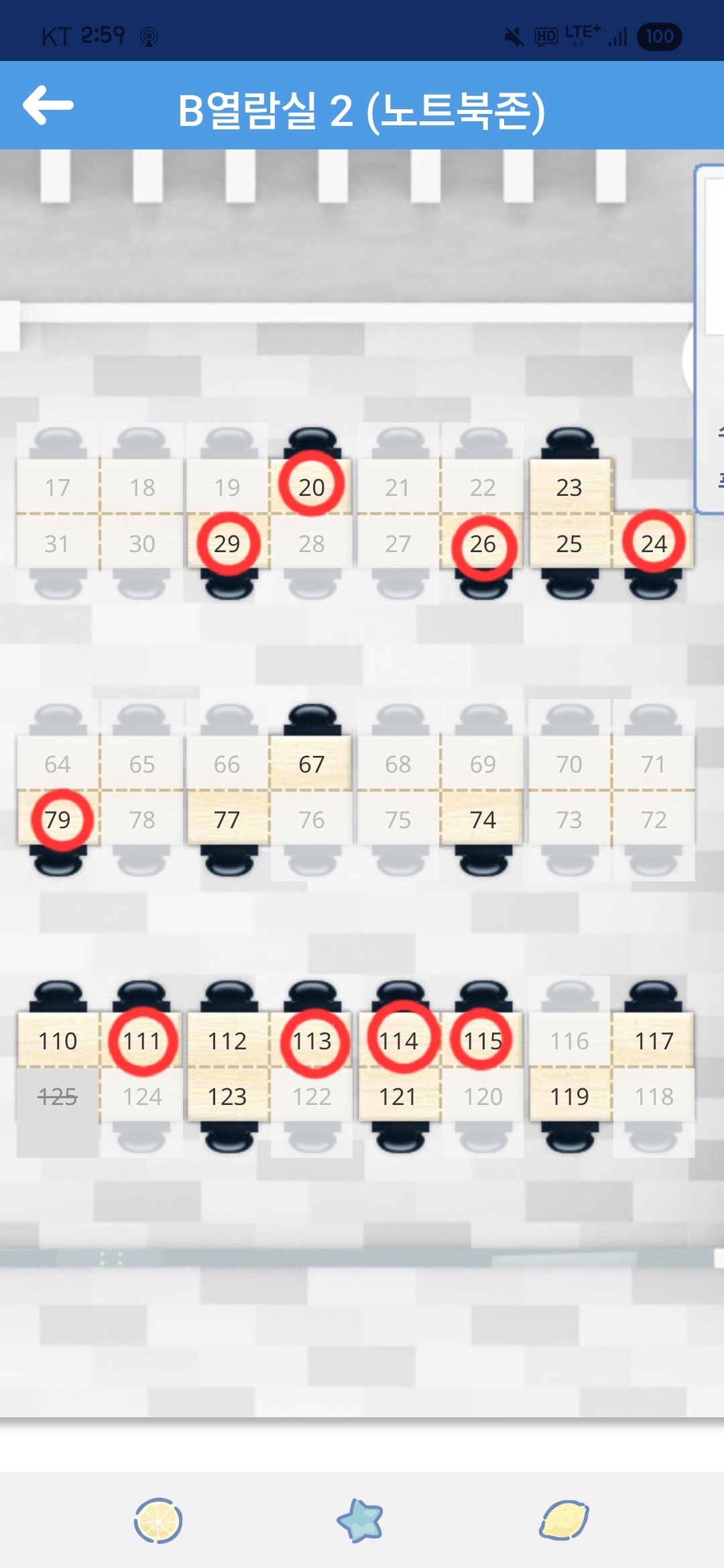


Figure 1. The SKKU Academic Information Center app UI and the seat currently in use without being checked in (red circle)

.

**ABSTRACT**

When using Dido (Samsung Library), it is possible to check seat availability through the SKKU Academic Information Center app. (Refer the figure 1) However, since many users occupy seats without officially checking in, the app has limitations in accurately reflecting the actual seat status.

To address this issue, our team project proposed a simple linear regression model and a GCN + GRU model that predicts seat occupancy by analyzing spatial and temporal correlations. Through this project, we achieved predictions that closely approximate the actual seat usage data. The significance of our approach lies in the attempt to model inter-seat correlations using a Graph Neural Network (GNN) framework.

**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

During exam periods, it is often difficult to find available seats at Dido (Samsung Library). Although the SKKU Academic Information Center app allows users to check seat availability in advance, the presence of many users who occupy seats without officially checking in limits the app's accuracy in reflecting the real-time seat status.

This project aims to predict the current seat occupancy by using date and seat information from the Academic Information Center as input. As a result, students will be able to get a rough estimate of seat availability—even when seats are used without being checked in—and thus use the library more efficiently.

Additionally, by providing information on seat availability by time of day, the system can offer tailored guidance to students with different schedules and purposes for using the library.

This report introduces two approaches to predicting actual seat occupancy: a regression-based method and a GNN-based method. In particular, the GNN approach combines GCN (Graph Convolutional Network) and GRU (Gated Recurrent Unit) to fully utilize both spatial and temporal information, which is a key novelty of this project.

# Related Work

In the current SKKU Academic Information Center app, as shown on the left side of Figure 1, users can view a seating layout along with the check-in status of each seat. If a seat is shown in light yellow, it is available for check-in; if it appears white, it has already been checked in.

While the existing app does indicate whether a seat has been officially checked in, it does not reflect the actual usage of seats. In practice, many users occupy seats without going through the official check-in process, creating a significant gap between the app's displayed check-in status and the real seat usage.

This discrepancy is illustrated on the right side of Figure 1. The red circles highlight seats that are physically occupied despite not being checked in.

Users who rely solely on the SKKU app to find available seats may face inconvenience due to this mismatch between the check-in data and actual seat occupancy.

This team project aims to address this issue by predicting the real-time seat usage through machine learning models. Specifically, it seeks to bridge the gap between the official check-in data and actual seat occupancy by learning from patterns and correlations in the data.

# Methodology

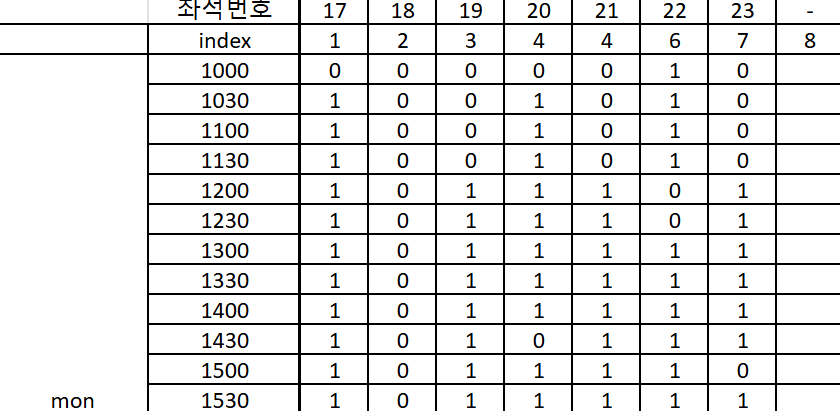
## Data Collection

|  |  |  |
| --- | --- | --- |
|  | **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

.



The information available from the existing SKKU Academic Information Center app corresponds to (a)+(c) and (b)+(d) in Table 1, while the target information to be predicted in this project is (d). Therefore, during the data collection process, we captured screenshots of the app to record either (b) or (d), since (d) can be derived when (a)+(c) and either (b) or (d) are provided.

Furthermore, we determined that recording only the total number of users rather than the occupancy status of each individual seat would be unsuitable for a GNN model, which requires node-level granularity. Thus, as shown in Figure 2, we recorded the usage status of every seat individually at each time interval. In this representation, a value of \*\*1\*\* indicates an unavailable seat (i.e., corresponding to (a), (b), or (c)), while a value of \*\*0\*\* indicates an available seat (i.e., corresponding to (d)).

### Data Collection Details

All data was collected from Reading Room 2, Section B, on Basement Level 1 of the Samsung Library.

Data was collected twice per day—between 10 AM and 1 PM (morning session) and between 1 PM and 4 PM (afternoon session)—at 30-minute intervals, resulting in 12 data entries per day.

Data collection was conducted over 12 days, from May 26 to June 6, and a total of 280 seat status entries were gathered.

### Data Collection Protocol

1. Take a screenshot of the seat status screen from the Academic Information Center app.

2. Walk around the reading room and count the number of people sitting in seats that have not been officially checked in.

3. Actual number of occupied seats = Number of checked-in seats + Number of users in unchecked-in seats.

## Data Augmentation

[추후 작성]

## Regression Model

Using the collected data and time information, a regression model was implemented. The date data was converted into two features: whether it was during the exam period and the day of the week. The exam period feature was arbitrarily set with May 28 as the cutoff date; dates before May 28 were labeled 0 (non-exam period), and dates from May 28 onward were labeled 1 (exam period). The day-of-week feature was assigned sequential values from 1 to 5 for Monday through Friday.

The model inputs included the exam period indicator, day-of-week feature, and the number of checked-in seats according to the app (i.e., the count of seats in states (a) or (c) in Table 1). The output was the predicted number of seats actually in use.

Thus, the model predicts the real congestion level of the space by using the current date and time data along with the check-in status provided by the SKKU Academic Information Center app.

### Model Structure

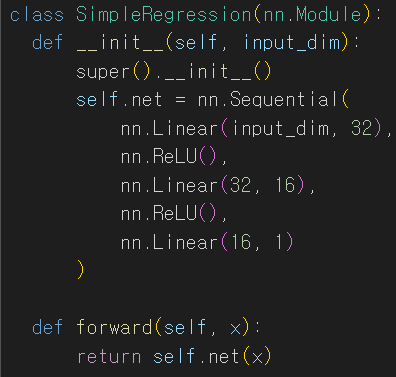


Figure 3. Regression model structure

.

Finally, we adopted the model structure shown in Figure 3. It is a regression model based on a Multi-Layer Perceptron implemented in PyTorch. The model takes an input with a dimension equal to input\\_dim and outputs a single-dimensional value. It consists of three Linear Layers, with forward propagation passing sequentially through layers of 32, 16, and 1 dimension.

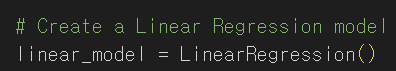


Figure 4. The structure of the linear egression model for comparison

.

As a baseline for evaluating the performance of our proposed model (Figure 3), we used the LinearRegression model from the sklearn library. (See Figure 4 for reference)

### Loss Function and Hyperparameters

We used Mean Squared Error (MSE) as the loss function, set the learning rate to 0.001, and trained the model for 10,000 epochs. Adam was used as the optimizer.

### Limitation

This regression model handles only the total number of seats without considering the status of individual seats, so it falls short of fully meeting user needs. Generally, when choosing seats, users tend to prefer sparsely occupied areas over densely crowded ones. However, this regression model cannot capture or reflect such spatial preferences, which is a significant limitation.

## GNN Model (GCN + GRU)

[장섭님 부탁드립니다..]

# Experiments

## Regression Model

Place Tables/Figures/Images in text as close to the reference as possible (see Figure 1). It may extend across both columns to a maximum width of 17.78 cm (7”).

Captions should be Times New

# Results

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.

## Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

# Conclusion

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.16147>.
2. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
3. Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= <http://doi.acm.org/10.1145/332040.332491>.
4. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
5. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
6. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.
7. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= <http://doi.acm.org/10.1145/964696.964697>.
8. Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= <http://dx.doi.org/10.1016/j.jss.2005.05.030>.
9. Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= <http://doi.acm.org/10.1145/90417.90738>.