

# Kingo Manager - Information Recommendation Application For University Students

Junwoo Kim, Shingyu Kang, Chunghyeon Park, Juhyeon Park

Sungkyunkwan University SWE3028\_41  
[https://github.com/rlawnsdn/SWE3028\\_41\\_team\\_H](https://github.com/rlawnsdn/SWE3028_41_team_H)

**Abstract.** 대학 생활에서 진로활동은 학업과 마찬가지로 취업과 졸업 이후의 진학 등에 큰 영향을 끼치기 때문에, 적절한 시기에 자신의 상황에 맞는 활동들을 하는 것이 중요하다. 하지만, 진로활동 정보들은 서로 다른 웹페이지들에 게시되며, 학업과 병행하여 자신과 맞는 정보를 찾는 일에는 많은 시간과 노력이 요구된다. 누군가 이 번거로움을 털어준다면 대학생들의 진로적인 부담을 크게 줄여줄 수 있으며, Kingo Manager는 이처럼 대학생들이 정보를 찾는 과정에서 겪는 번거로움을 해소해주는 진로활동 추천 서비스이다. Kingo Manager는 웹 크롤링 방식을 통해서 다양한 사이트에서 필요한 정보를 수집하고, 기계학습 모델 중 N-GCN을 기반으로 한 추천시스템을 이용하여 본인에게 맞는 추천과 목, 동아리 모집공고, 학술대회, 채용공고 등의 정보를 제공해준다. 또한 모델 학습을 통해 각 학생들의 학년, 관심분야를 고려하여 개인별로 알맞은 정보를 추천해 준다. 추천 받은 정보들은 메신저 형태의 어플리케이션을 통해서 확인할 수 있고, 해당 정보들은 언제든지 다시 검색하여 찾아볼 수 있다. 어플리케이션 구현에서 프론트엔드는 React-Native를 사용하였고, 백엔드는 Flask 와 AWS 서비스를 사용하였다. 추천 알고리즘은 사용자 관심도를 바탕으로 학습되기 때문에, 사용자 맞춤 컨텐츠를 제공할 수 있으며, 현재는 성균관대학교 소프트웨어학과 학생들만을 대상으로 하는 어플을 개발하였다.

**Keywords:** N-GCN(Multiscale Graph Convolutional Networks), Recommendation System, React-Native and AWS based Application

## 1 Introduction

### 1.1 Statement

대학생활은 오로지 학업에만 열중하였던 고등학교와 달리 자신의 진로활동에 필요한 정보를 직접 찾아봐야 한다. 하지만 학기 중에는 과제나 프로젝트 등의 학업과 병행하여 시기에 적절한 진로활동들을 찾아보기가 쉽지 않다. 또한 진로활동 정보는 여러 홈페이지에 널리 퍼져 있기 때문에, 본인에게 맞는 활동이 무엇인지 찾는데 많은 시간을 소비하게 되며, 적극적으로 홍보를 하지 않는 홈페이지의 공지사항들은 확인하기가 어렵다. 한 예시로 'UR1004'는 학교에서 제공하는 봉사활동 주선 사이트인데, 거의 홍보되지 않아서 학생들이 잘 모르며, 학생들은 필요한 정보가 있으면 성균관대학교 GLS 포털, 산학협력 톨게이트, 캘린지스퀘어, 학생성공게이트와 같은 다수의 홈페이지를 찾아 보아야 했다. 이처럼 많은 학생들은 강도높은 학업과 병행하여 진로활동을 찾아보는데에 조차 번거로움을 겪고 있다.

## 1.2 Motivation

성균관대학교에만 존재하는 특별한 졸업요건으로서 3품 인증이 있다. 그 중 창의품을 따는 방법에는 여러가지가 있지만, 가장 간단한 방법은 'Etest'에 응시하는 것이다. 우리 프로젝트의 팀원은 모두 졸업을 앞둔 고학년으로 이루어졌지만, 창의품을 따는 방법 중에 Etest가 가장 쉽다는 사실을 아는 사람은 거의 없었다. 이러한 정보를 아는 방법은 학교 생활을 하면서 선배, 혹은 동기에게 듣는 것 뿐이고, 이는 많은 대학생들이 대부분의 정보를 스스로 찾아야 하는 경우가 많은 대표적인 사례이다. 많은 대학생들이 오리엔테이션 시간이나 Kingo-M, 에브리타임, Freshman Guide를 통해서 정보를 얻게 되는데, 오리엔테이션 시간은 일회성이고, 에브리타임은 학교 외부의 어플리케이션이므로 필요한 정보를 얻기가 쉽지 않다. 또한 학사 공지사항 등을 전달해주는 Kingo-M의 메신저 기능도 개개인에게 맞춤화된 알람이 아닌, 일괄적인 정보를 전달해주는 형태이기 때문에, 사용자는 자신과 관련이 없는 정보들도 읽어야 하는 한계점이 존재한다. 특히 코로나 시기로 선후배간 연계가 끊어진 점, 복수전공생에게는 선후배를 만나기 쉽지 않은 점 등을 고려해보면 구전으로 전해지는 정보들에 대해서 명문화의 필요성이 느껴졌고, 이를 위해서 Kingo Manager 프로젝트를 착수하게 되었다.

## 1.3 Proposal and Goal

이 프로젝트에서는 설문조사를 통해서 학생들에게 필요하지만 많이 알지 못하는 사이트를 발굴하고, 학년에 맞는 꼭 필요한 정보를 추린 후 N-GCN 모델을 통해서 사용자에게 맞는 정보를 학습시킨 후 학생들에게 필요한 정보를 제공하는 것을 목표로 한다. 다양한 사이트에서 필요한 정보만 가져오는 것을 목적으로 하며, 사용자가 받는 정보들은 누적되고, 검색기능을 통해서도 찾을 수 있으며, 사용자의 클릭 정보는 서버를 통해 Local에 저장되어서 모델을 재학습시키는 방법으로 사용자에게 최적화된 정보를 제공하는 것을 목적으로 한다.

## 1.4 Technique and Approach

우선 설문조사를 통해서 제공할 정보의 바운더리를 설정하고 알려져 있지 않은 유용한 사이트들을 발굴한다. 조사 결과는 두 가지 목적으로 사용되는데 첫번째로 설문조사 결과를 Augmentation하여 Machine Learning(이하 ML) Model을 학습시키기 위한 데이터셋을 만들고, 두 번째로 발굴된 사이트들과, 기존에 알려진 사이트들을 참고하여 Web Crawling의 범위를 설정하고 정보를 모아서 제공한다. 이렇게 얻어낸 정보와 Dataset은 AWS 서버에서 받아온 사용자의 정보를 토대로 N-GCN을 거쳐서 추천정보 Top 3를 선정하는 데 사용된다. ML을 학습시키기 위한 dataset과 크롤링 결과는 Local에 저장되며, 서버에서는 사용자별 학년과 관심사, 그리고 특정 정보에 대한 방문 횟수를 받아서 ML을 학습시키기 위한 데이터로 변환하고 저장한다. 매일 자정에 ML Training, Web Crawling 하여서 정보를 업데이트하고, 고객의 관심사를 반영할 수 있도록 한다.

## 1.5 Requirements

아래의 ??은 어플리케이션을 구현하기에 앞서서 필요한 기능들을 정리한 requirements 테이블이다. 테이블을 참고하면 requirements 번호와 구현하기에 필요한 component가 정리되어 있음을 알 수 있다.

No.	분야	수행내용
1	Frontend	회원가입/로그인
2		메인페이지
3		검색기능
4		마이페이지(회원 정보 관리)
5	Backend	회원가입/로그인 및 이메일 전송 서버 구축
6		aws Lambda와 RDS를 연동한 추천 컨텐츠 전송 시스템 구축
7		사용자 관심도를 반영한 AI Model 학습 파라미터 관리 서버 구축
8		서버 배포 및 시스템 유지
7	Data	설문조사
8		Data Boundary 설정
9		Data Augmentation 및 수정
10		Data Crawling
11	AI	모델 설정
12		모델 input,output 수정
13		사용자 클릭 모델 반영 및 재학습 시스템
14		Backend 연동
15		AI 학습 및 Backend post 자동화 시스템

**Table 1.** Requirements.

## 1.6 Achievement and empirical Evaluations

기존의 제시한 requirements 들을 모두 만족하기는 했으나, 몇가지 제약사항이 있으며, 그것은 추후 Limitation and Discussions 에서 설명하도록 하겠다.

## 2 Related Work

### 2.1 Kingo-M

성균관대학교의 학생들을 대상으로 학교 행사 혹은 정보제공을 해주는 시스템은 Kingo-M(Kingo Mobile)의 메시지함 기능이 거의 유일하다. 해당 메시지함은 학교 차원에서 홍보하고자 하는 공지사항 혹은 학생들이 시기적으로 읽을 필요가 있는 자료를 모바일에 메시지 형식으로 제공한다. 이는 학생들에게 유용한 정보를 모바일이라는 가독성이 좋은 매체를 통해서 전송해주는 관점에서 우리 프로젝트가 구상하는 아이템과 유사하다. 하지만 프로젝트의 모든 팀원이 느끼는 Kingo-M의 메시지함 기능에 대한 인상은 자신과 별로 상관없는 정보가 제공되어서 메시지를 거의 읽지 않게 되는 것이었다. Kingo Manager는 사용자의 학년과 3품 정보 등의 학사 정보를 수집하고, 자신의 시기와 관심사에 맞는 자료들을 추천시스템을 이용하여 선정하여 제공한다. 대학을 졸업하기까지 학생들은 높은 강도의 학업을 병행하면서 자신의 시기에 맞는 진로활동을 해야하며, 여기에 많은 시간과 노력이 필요하게 된다. Kingo Manager는 이처럼 학생들이 진로활동을 찾아보아야 하는 번거로움이 존재함을 인식하여 개선하려고 하며, 추천 시스템을 통해서 Kingo-M보다 나은 수준의 자료를 제공할 수 있다.

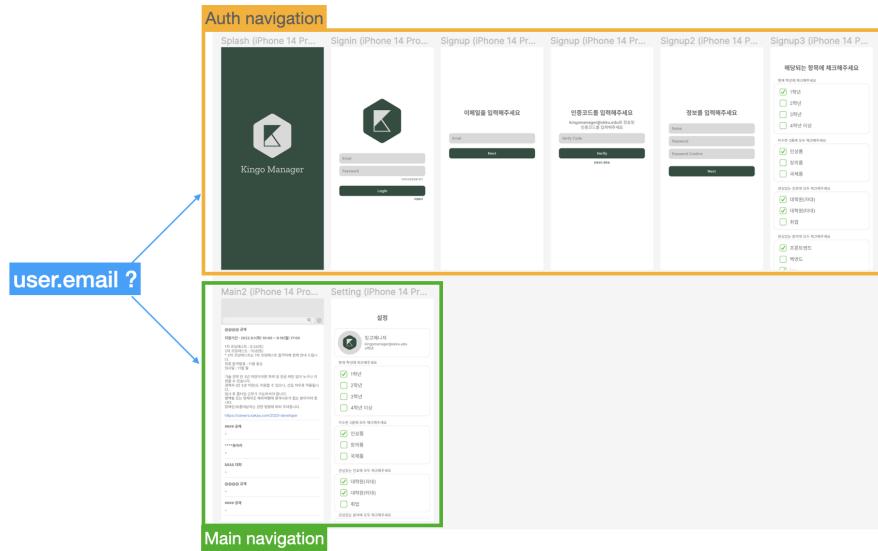
## 2.2 GNN(Graph Neural Network)

인터넷과 광고, E-commerce가 발달하면서 다양한 추천 알고리즘이 소개되고 발달하고 있다. 그중에 인터넷에서 대표적으로 사용되는 알고리즘으로는 실시간으로 Session에서 주어지는 정보를 통해 사용자의 general interests와 current interests를 알아내는 방식의 'Session-based Recommendation with Graph Neural Networks' 가 있다.[1] 이 모델은 실시간으로 웹 페이지를 사용하고 있는 사용자의 Session 정보를 이용해서 general interests 와 current interests를 구하는 것에 초점을 두고 있다. 우리 프로젝트에서도 N-GCN 기반의 추천 알고리즘을 설계하였으며, 사용자의 컨텐츠 열람 횟수를 체크하여 모델의 학습 파라미터를 생성하는 방식으로 추천 알고리즘을 기획하였다.

## 3 Design

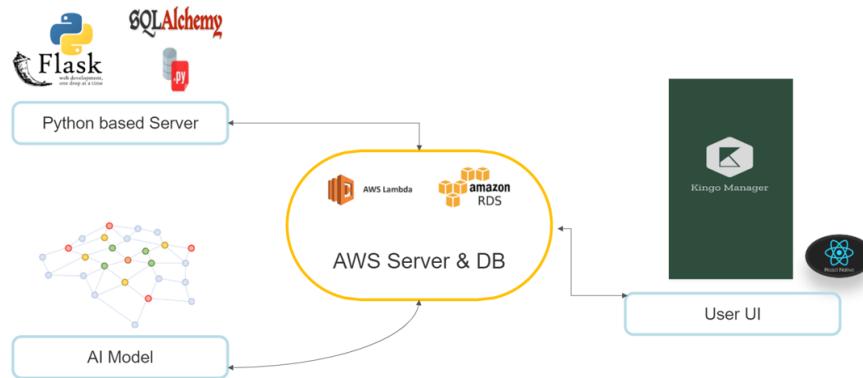
### 3.1 Overall Architecture

**3.1.1 Frontend architecture** Kingo Manager 어플리케이션의 프론트엔드는 Auth navigation과 Main navigation으로 이루어져 있다. Auth navigation은 Login, Signup pages가 포함되어 있고, Main navigation은 Main, Settings page로 이루어져 있다. useContext를 이용하여 만든 전역변수 user의 email값이 null이면 Auth navigation을 띠우고, 값이 존재한다면 Main navigation을 띠워준다. 전체 프론트엔드 아키텍처를 도식화한 그림은 ??을 참고.



**Fig. 1.** frontend architecture

**3.1.2 Backend architecture** Kingo Manager의 전체적인 동작은 사용자가 자신의 관심사에 맞는 추천정보를 얻는 것과, 사용자의 관심사를 추천 알고리즘에 반영하여 모델을 학습시키는 과정을 통해서 이루어진다. 즉, AI Model과 사용자 UI가 양방향으로 필요한 정보를 통신할 수 있도록 백엔드 아키텍처를 설계하였고, 추천정보는 Local PC로부터 데이터베이스에 post 요청을 통해서 저장된 후, 다시한번 서버로부터 어플리케이션에 전송되는 방식으로 구현되었다. 또한 학습 파라미터를 전송하는 구조는 반대의 방향으로 이루어지는데, 어플리케이션에서 사용자의 컨텐츠 열람 횟수를 누적하여 카운트한 후에 100회 이상 횟수가 누적되면, 어플리케이션에서 서버로의 post 요청이 발생한다. 사용자 관심도 정보는 데이터베이스에 저장되며, Local PC가 필요한 때에 서버로부터 PC로 전송된다. 전체 백엔드 아키텍처를 도식화한 그림은 ??을 참고.



**Fig. 2.** backend architecture

### 3.2 Core Technique

프론트엔드에서는 React-native를 사용하여 어플을 개발하였다. 어플의 구조가 복잡한 편이 아니고, 어플 개발에 사용되는 인력이 적은 만큼 ios와 android를 동시에 개발 할 수 있는 크로스 플랫폼 앱이 적절하다고 판단하여 react-native를 선택하게 되었다. 백엔드와의 통신을 위해서 Promise기반 HTTP를라이언트인 axios 사용하였고, 로컬에서 사용자의 정보를 저장하기 위해서 react-native AsyncStorage를 사용하였다.

백엔드 서버는 전반적으로 AWS(Amazon Web Services) 의 cloud service를 사용하였고, 소스코드는 python 기반의 Flask API와 SQL Alchemy 라이브러리를 적극적으로 활용하였다. AWS에서는 서버를 설계하기 위한 거의 모든 기능을 제공하는데, 우리 프로젝트에서는 Lambda(Server-less Computing Services) 와 RDS(Relational Database Storage) 를 적극적으로 활용하였다. Lambda 는 Event-Driven-Action 이 가능하기 때문에 추천 정보와 학습 파라미터를 원하는 시간에 송수신하기 위한 적절한 서비스로 여겨졌다. 프로젝트에서 필요로하는 3종 정보와 같은 사용자 정보, 추천정보, 학습

파라미터를 효과적으로 관리 및 가공하여 사용하기 위해서 관계형 데이터베이스인 RDS를 사용하였고, Lambda 함수로부터 특정 기능의 데이터에 대한 post가 서버에 발생했을 때 RDS에 존재하는 데이터를 적절히 조합하여 서버의 응답값을 줄 수 있었다. 추가적으로, 서버의 소스코드를 aws Lambda에 배포하고, 업데이트 하는 과정에서 zappa 라이브러리를 사용하였다.

### 3.3 Challenges and Overcoming

프론트엔드 파트에서의 Challenge는 사용자들이 사용하기 편리하고 직관적으로 받아들일 수 있는 메인 페이지의 구성방식을 결정하는 것이었다. 우선, 에브리타임같은 방식의 게시물을 클릭해서 다른 페이지로 넘어가 게시물의 본문을 보여주는 방식은 배제하기로 하였다. 이러한 방식은 모든 게시물을 일일히 클릭해서 들어가야 본문을 읽을 수 있고 다시 메인 페이지로 나오기 위해서 뒤로가는 버튼을 한번 더 눌러야 하는 수고가 있기 때문에 여러가지 정보를 한번에 사용자에게 보여주기 원하는 우리 프로젝트의 목표와 맞지 않다고 생각하였다. 그래서 두번째로 생각한 방식은 카카오톡의 메세지처럼 모든 게시물의 본문을 메인 페이지에 전부 보여주는 방식을 고안하였는데, 이 방식의 문제점은 만약 현재 스크린에 두 개 이상의 게시물이 렌더링되어 있으면 사용자가 어떤 게시물을 보고 있는지 알아낼 수 없다는 점이었다. 이는 사용자가 어느 게시물들을 많이 보는지에 대한 정보를 취합하여 다시 추천 시스템에 적용하기는 우리 프로젝트에서는 선택하기 어려운 방식이었다. 그래서 최종적으로 사용자가 게시물을 보려면 한번의 클릭을 필요로 하지만 적어도 뒤로가기 버튼은 누르지 않아도 되고, 사용자가 어떤 게시물을 선호하는지 명확하게 구분할 수 있는 Flatlist에서 게시물을 클릭시 그 component만 re-rendering하여 본문을 보여주는 방식을 채용하게 되었다.

백엔드 파트에서는 서버를 설계하면서 한글 엔코딩 및 디코딩을 지원하기 위해서 많은 어려움이 있었다. 팀원이 사용하는 컴퓨터의 OS가 윈도우, MacOS로 나뉘었고, 서버에서 응답값으로 준 json 데이터가 윈도우 환경에서는 깨지지 않고 보이지만, MacOS에서는 깨져서 알아볼 수 없는 등의 문제가 발생했다. 또한 데이터베이스에 추천정보 및 사용자 정보 등을 등록하는 과정에서도 한글에 일관적으로 'utf-8' 엔코딩 및 디코딩을 적용해 주어야 했다. 이를 해결하기 위해서는 문자열 변수를 다룰 때마다 서버 내부적으로 post 값에는 encoding, request 값에는 decoding이 적용되도록 디버깅하였다. 이 과정에서 python 라이브러리인 urllib의 parse.quote과 parse.unquote을 주로 활용하였다.

AI 파트에서는 두 가지의 Challenges가 있었는데 첫번째로 설문조사를 통해서 feature 와 ground truth 값을 뽑아낼 수 있도록 설문조사 질문을 작성하는 것이 어려웠다. 먼저 무엇을 feature로 삼을지 정해야했고, 무엇을 ground truth로 잡아야 할지 정해야 했다. 고심끝에 feature로는 대학생 개개인에게 필요한 정보의 종류를 다르게 만드는 요인중 가장 큰 요인인 학년과, 개개인의 관심사를 feature로 설정했고, ground truth 값은 feature를 가진 개인에게 어떠한 정보를 추천해주는 것이 좋을지에 대한 결과이므로 선배들에게 후배에게 필요한 정보가 무엇인지 묻는 것을 통해서 얻었다. 두번째로 데이터셋에 맞는 모델을 선정하고 전처리를 원활하게 해주는 것이 어려웠다. 결국 많은 실패를 하고 시간을 들여 찾으면서, 데이터 전처리의 형태랑 모델을 선정할 수 있었다.

### 3.4 Reason about Model Selection

사용자와 카테고리나 콘텐츠 간의 패턴이 중요한 것이 아니라 관계가 중요한 부분이므로, 패턴만 가지고 결과를 추측하는 모델이 아니라 각 항목마다의 관계를 가지고 결과를 추측하는 모델이 필요했다. 그래서 GNN 관련 모델을 선정하게 되었다. 그리고 GNN 관련 모델들 중에 데이터셋의 관계를 잘 나타낼 수 있는 모델을 선정했다.

### 3.5 ML model - Build Dataset

설문조사를 통해서 얻어야 하는 것은 총 3가지였다. 첫째로 대학생들에게 도움이 되는 정보, 사이트들을 발굴해야 했고, 둘째로 학년별로 관심사가 무엇인지 알아내야 했다. 셋째로 선배의 입장으로서 학년별 후배들에게 추천하는 정보들에 대한 정보를 모아야했다. 따라서 설문조사지를 다음 ??와 같이 제작, 배포했으며, 스타벅스 아이스아메리카노를 보상으로 걸어서 설문조사를 장려했다. 이렇게 발굴한 사이트에서 Web Crawling 기술중 BeautifulSoup4와 Selenium 기술을 활용하여 데이터를 모았다.

1	전화번호가 어떻게 되시나요?
2	지금 학기수가 어떻게 되시나요?
3	소프트웨어 원전공생이십니까?
4	구3품과 신3품중 어느쪽 대상자이십니까?
5	인성품을 어떤 것으로 취득하셨나요? 혹은 하실 예정이신가요?
6	국제품을 어떤 것으로 취득하셨나요? 혹은 하실 예정이신가요?
7	창의품을 어떤 것으로 취득하셨나요? 혹은 하실 예정이신가요?
8	AI품을 어떤 것으로 취득하셨나요? 혹은 하실 예정이신가요?
9	인턴십인증을 어떤 것으로 취득하셨나요? 혹은 하실 예정이신가요?
10	3품 면제자 이십니까?
11	학사/취업/진로 활동에 대한 정보를 얻기위해 자주 방문하는 페이지가 있나요?,
12	1학년에게 가장 추천하는 활동은 무엇인가요? (최대 3개)
13	2학년에게 가장 추천하는 활동은 무엇인가요? (최대 3개)
14	3학년에게 가장 추천하는 활동은 무엇인가요? (최대 3개)
15	4학년에게 가장 추천하는 활동은 무엇인가요? (최대 3개)
16	지금 당신이 가장 관심있는 활동(혹은 분야)이 무엇입니까? (최대 3개)
17	마지막 질문입니다. 이후 진로 희망이 어떻게 되시나요?(최대 3개)

Table 2. Survey.

### 3.6 ML model - N-GCN(Multi-Scale Graph Convolution Network)

Graph는 vertices(node)와 edge로 구성된 구조를 말한다.

$$G = (V, E) \quad (1)$$

(1)과 같은 식으로 나타낼 수 있는데, 각 node는 feature vector를 가지고 있다. 앞에서 말한 것처럼, 기존의 모델과 다르게 그래프를 사용해서 얻을 수 있는 차별점으로는

노드들 사이의 패턴을 찾는 데 집중하는 것이 아니라 노드들 사이의 관계에 집중한다는 것이다. Adjacency matrix(A)와 각 feature vector로 구성된 feature matrix(H)를 통해 GNN layer를 통과할 때마다 각 matrix(H)의 정보를 업데이트 해준다.

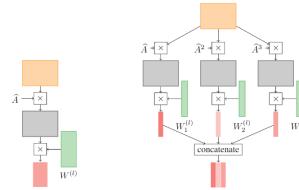
$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)} + b^{(l)}) \quad (2)$$

추가로, Adjacency matrix에서 각 노드에 연결된 가중치는 서로 다를 것이므로, Node embedding을 통해 구한 값인

$$z_u^T z_v \quad (3)$$

를 통해 각 노드들마다 Pagerank를 사용해 각 노드 사이에 어느 정도 연관성이 있느지를 가중치를 통해 나타낸다. 이 Pagerank를 사용해서 GNN layer를 지날 때마다 가중치를 최신화 시켜 각 노드의 가중치가 수렴을 하게 된다. GCN을 사용했기 때문에, 여기에 추가적으로 Convolution을 적용해 H를 구해 나간다.

최종적으로, Figure2와 같이 사용한 모델인 N-GCN은 위의 GCN의 과정을 사용하는데, GCN 결과를 층(layer)을 나눠서 결과를 계산하고 그 결과들을 Concatenate해서 최종 결과를 도출한다.



**Fig. 3. Model <sup>1</sup>**

## 4 Implementation

### 4.1 UL/UX

추후에 다른 학교, 다른 과 학생들을 대상으로도 확장할 가능성을 염두에 두 프로젝트이기는 하지만 우선 현재의 목표는 성균관대학교 소프트웨어학과 학생들에게 도움이 되는 서비스를 제공하는 것이므로 디자인에는 성균관대학교 공식색 중 하나인 진초록: #2E4E3F를 사용하였다. 그 외에 사용한 색으로는 연한 회색: #D9D9D9, 진한 회색: #6C6C6C, 검정: #000000, 하양: #FFFFFF을 사용하였고, 경고 메세지를 위해 빨강: #E84118, 하이퍼링크 강조를 위한 파랑: #4472C4를 사용하였다.

**4.1.1 SplashScreen** 처음 어플을 실행시키면 나오는 어플을 로딩할 때 나오는 창이다. 정 가운데에 Kingo Manager의 로고가 박혀있으며 배경은 성균관대 공식색 중 하나인 #2E4E3F로 되어있다. ?? 참조.

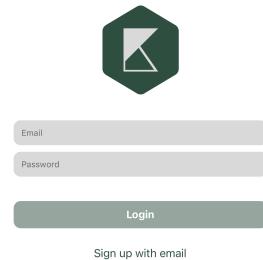
---

<sup>1</sup> Image is from A Higher-Order Graph Convolutional Layer. Sami A Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Hravir Harutyunyan. NeurIPS, 2018.



**Fig. 4.** SplashScreen

7:34      [Battery icon]    [Signal icon]



**Fig. 5.** Login page

**4.1.2 Login page** 가운데 상단에 kingomanager\_logo\_green을 배치하였고 그 밑에 이메일을 입력하는 창과 비밀번호를 입력하는 창을 배치하였다. 두개의 창은 같은 컴포넌트로 구성되어 있으며, 각 창을 focus하면 검은색 테두리를 이용해 현재 사용자가 선택한 input component가 무엇인지 보여준다. 또한 email을 입력하고 난 후에는 자동으로 password 입력칸으로 focus가 옮겨가도록 하였다. 또한 password 입력칸에 입력되는 문자열은 보안을 위해 점으로 표시된다. ?? 참조.

**4.1.3 Signup pages(1~ 6)** 모든 Signup.pages에서는 사용자에게 텍스트를 입력 받는 input 컴포넌트와 다음 페이지로 넘어가는 button 컴포넌트를 재사용한다. Input 컴포넌트의 경우 isFocused 효과를 넣어서 사용자가 input을 클릭하면 테두리의 색을 검은색으로 변경하며 사용자에게 현재 입력받고 있는 창을 보여준다. 또한 Signup\_page3같이 한번에 여러 개의 입력값을 받아야 하는경우 입력값을 입력한 후에 Enter키를 누르면 다음 input으로 자동으로 focus가 넘어가도록 설정하였다. 이로 인해 사용자는 일일이 input component를 입력하지 않아도 되는 수고를 덜 수 있다. Signup\_page1 화면에서는 이메일을 입력받고, 이메일이 형식에 알맞은 이메일이라면 다음 페이지로 가는 버튼이 활성화된다. 그 후 서버에서 입력된 이메일이 다른 계정의 이메일과 중복되었는지를 판단한 후에 중복되지 않았다고 판단되면 입력된 이메일로 verification code를 전송 후 다음 페이지인 Signup\_page2로 넘어간다. Signup\_page2에서 사용자는 본인이 입력한 이메일로 들어가 verification code를 확인 후 입력해야 한다. 서버에서는 입력받은 verification이 일치하는지 확인 후 일치한다면 Signup\_page3로 넘어간다. Signup\_page3에서는 이름, 비밀번호, 비밀번호 확인을 입력해야 한다. 이름은 형식에 제한받지 않지만 아무것도 입력하지 않을 수는 없다. 비밀번호는 최소 8자리 이상이어야 한다. 또한 비밀번호와 비밀번호 확인 창에서는 입력된 값이 일치하는지 검증하여 일치한다면 다음 페이지로 넘어가는 버튼을 활성화시킨다. Signup\_page4~6에서는 체크박스를 이용해 사용자의 학년, 3품 인증 현황, 관심사를 입력받는다. 학년, 관심사의 경우 하나만 선택해야 하고 3품 인증 현황의 경우 중복 체크가 가능하다. ?? 참조.

**4.1.4 Main page** 로그인에 성공한다면 main\_page로 넘어가게 된다. main\_page에서는 사용자의 계정이 지금까지 추천받았던 게시물들을 보여준다. 추천 게시물들을 React native의 Flatlist를 이용하여 보여주고 가장 최근에 추천받은 게시물이 맨 위에 stack처럼 쌓이게 된다. 각각의 게시물들은 제목과 카테고리를 보여주고 사용자가 게시물을 클릭하면 해당 게시물만 다시 rendering하여 게시물의 본문과 하이퍼링크를 사용자에게 보여준다. 사용자는 선택한 게시물에서 하이퍼링크를 클릭해서 게시물과 관련된 웹페이지로 바로 넘어갈 수 있다. 또한 맨 위의 검색창을 이용하여 사용자가 찾는 게시물을 검색할 수 있다. 사용자가 검색어를 입력하면 입력받은 검색어가 게시물의 제목, 본문, 카테고리, 하이퍼링크 중 하나에 포함되어 있다면 해당되는 게시물들만 필터링하여 보여준다. main page에서 위로 드래그를 할 경우 Flatlist를 새로고침하여 추천된 새로운 게시물이 있다면 추가해준다. ??참조. 검색창의 오른쪽에 있는 x 버튼을 누르면 검색어를 한번에 삭제할 수 있다. 그리고 main.page의 헤더에 있는 톱니바퀴를 선택하면 Settings.page로 넘어가게 된다. ?? 참조.

**4.1.5 Settings page** Settings.page에서는 맨 위에 이름, 이메일, 학년이 표시된다. 그 밑에는 현재 계정에 설정되어 있는 학년, 3품 인증 현황, 관심사가 체크박스로 표시된다. 사용자는 체크박스에 체크된 항목을 변경한 후 save 버튼을 통해 본인 계정의

The figure consists of three screenshots of a mobile application's sign-up interface, showing the progression from basic information input to interest selection.

**Screenshot 1: Email Input**

Time: 7:34

Text: "이메일을 입력해주세요." (Please enter your email.)

Fields: "Email" (dorung00@naver.com) and "Verify Code" (dorung00@naver.com).

Buttons: "Next".

**Screenshot 2: Verification and Name Input**

Time: 7:36

Text: "나머지 정보를 입력해주세요." (Please enter the remaining information.)

Fields: "Name" (dorung00), "Password", and "Password Confirm".

Text: "Please enter your name." (Please enter your name.)

Buttons: "Next".

**Screenshot 3: Academic Year Selection and Interest Selection**

Time: 3:19

Text: "본인의 학년을 선택해 주십시오 (1/3)" (Please select your academic year (1/3))

Options: 1학년, 2학년, 3학년, 4학년 이상. The 4학년 이상 option is checked.

Text: "이미 획득한 상품이 있다면 체크해 주십시오 (2/3)" (If you have already obtained a product, check it (2/3))

Options: 인성품, 청의품, 국제품. The 인성품 and 청의품 options are checked.

Text: "본인의 관심사를 선택해 주십시오 (3/3)" (Please select your interests (3/3))

List of interests:

- 학점관리
- 자기이해와 진로탐색
- 봉사
- 동아리, 학회
- 외국어 점수 획득(국제급)
- 자격증
- 공모전/외부활동
- 졸업요건
- 기업인턴십/산학협력
- 취업캠프/서류, 면접 클리닉
- 대학원/학석연계

Buttons: "Next" and "Sign up".

Fig. 6. Signup pages

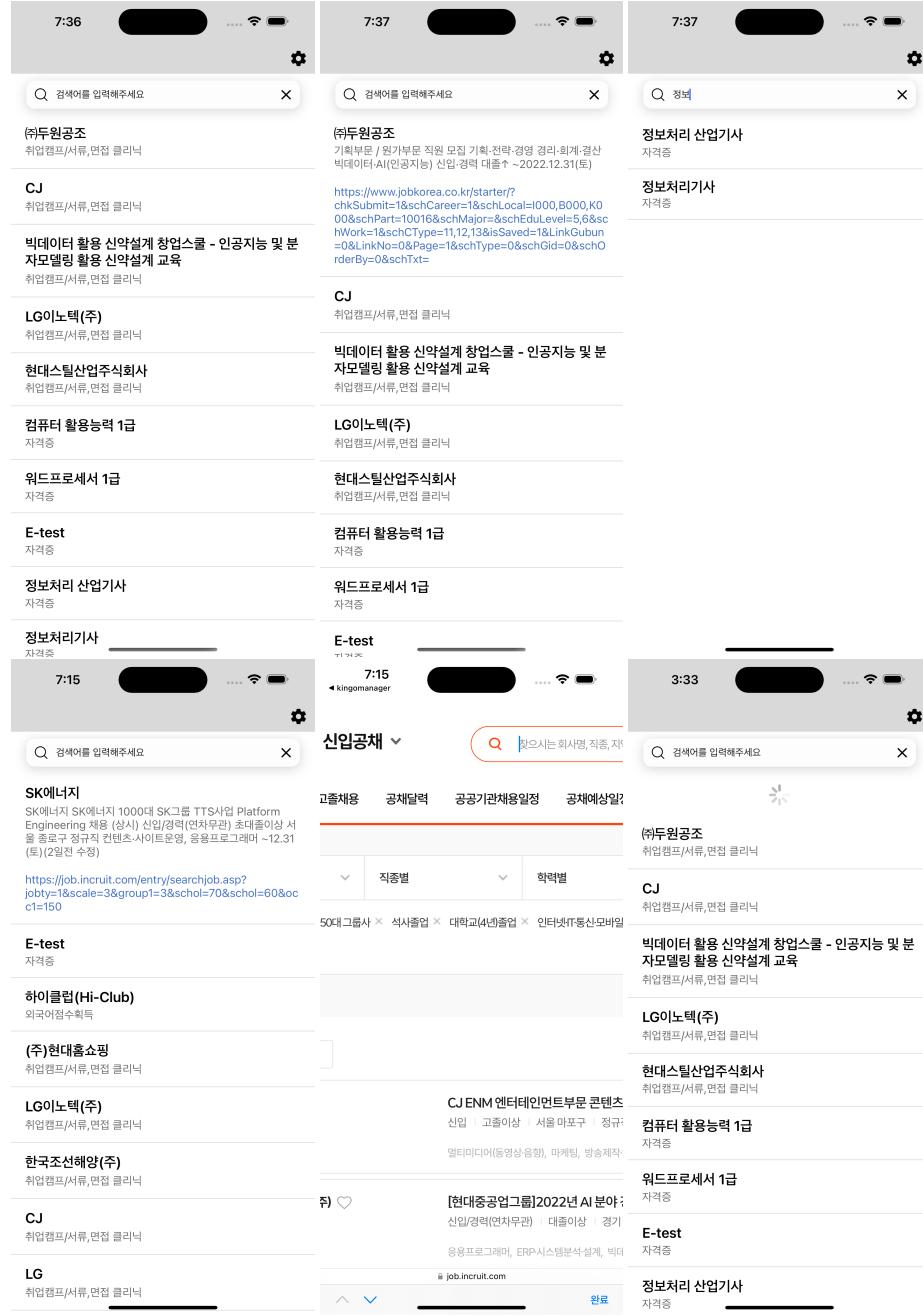


Fig. 7. Main pages

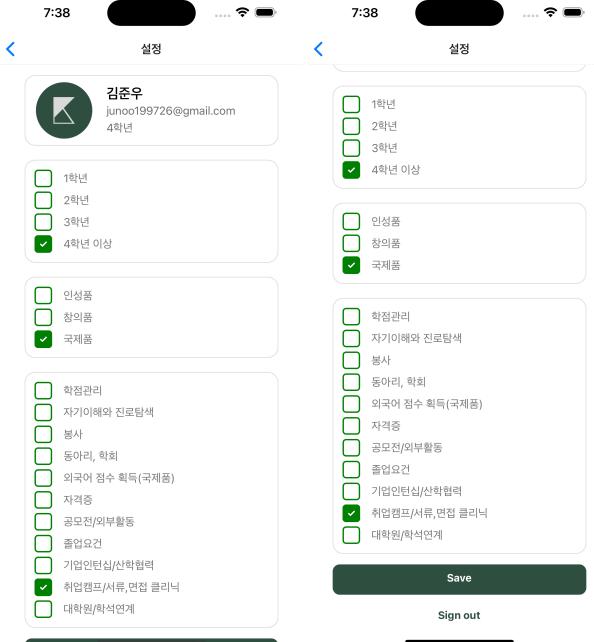


Fig. 8. Settings pages

설정값을 변경할 수 있다. 또한 sign out 버튼을 통해 로그아웃을 하여 Login\_page로 돌아갈 수 있다. ?? 참조.

또한 local storage를 사용할 수 있게 해주는 오픈소스인 react-native-Async-storage를 사용하여 사용자의 게시물 정보와 이메일, 카운트값, 3품 인증 여부, 학년, 관심사를 저장한다. Async-storage는 다음과 같은 key, value값으로 구성되어 있다. Table. ?? 참조.

#### 4.2 Server Functions and Model & UI communication

백엔드 서버가 제공해야하는 기능은 크게 3가지인데, 첫번째로, 기본적인 로그인 및 회원가입 기능과 회원가입시 사용자가 기재한 3품정보 및 관심 컨텐츠 등의 데이터를 서버상에 저장해야 한다. 이러한 사용자 정보는 AI Model의 feature vector로서 사용되기 때문에, 또한 AI 모델에서 서버를 통해 정보를 받을 수 있도록 양방향으로 설계하였다. 서버상에 사용자 정보를 저장하는 것은 aws RDS(Relationship Database Storage)에 해당 테이블을 생성한 후 데이터를 저장하는 방법으로 구현하였다. 사용자 정보를 저장하기 위해 설계한 테이블은 ??와 같고, aws Lambda와 연동하여 서버에 요청이 들어왔을 때 적절한 정보로 가공하여 응답해주도록 서버를 설계하였다. 다음으로 서버가 제공해야하는 기능은, AI Model로부터 얻은 사용자의 추천 정보를 사용자 UI에 전달하는 것이다. 앞서 설명한 RDS와 Lambda 기반의 서버를 이용하여 새로운 추천정보가 형성되면 이 정보의 대상이 되는 사용자 정보와 연관시켜서 추천정보를 RDS DB상에 저장하고, UI에 해당 정보를 보내주는 방식으로 서버를 구축하였다. 추천정보 테이블 구성을 ??와 같다.

key 값	value 값
'user_id'	유저의 이메일 값
'selectcount'	각 카테고리별로 클릭한 값의 배열
'grade'	학년
'insung'	인성품 취득 여부
'changeui'	창의품 취득 여부
'insung'	인성품 취득 여부
'0'	첫번째 게시물
'1'	두번째 게시물
'2'	세번째 게시물
...	...

**Table 3.** Async-storage

마지막으로, 서버는 사용자의 관심도 정보를 서버에 기록한 후, AI Model이 사용자 관심도 정보를 정기적으로 학습할 수 있도록 데이터를 제공하여야 한다. 이 기능은 UI에서 사용자가 특정 컨텐츠를 열람할 때마다 카테고리당 열람 횟수를 누적시킨 후에, 100번 이상의 열람이 일어날 때마다 서버상에 해당 데이터를 저장시킨 후에, AI Model에서 이 값을 받아와서 학습시키는 방법으로 구현되었다. 관심도 정보를 담은 테이블은 ??와 같이 구성되었다.

id(primary)	email	password	name	grade	insung	changeui	gukje	interest
1	capstone@gmail.com	capstone	캡스톤	4	True	True	False	10

**Table 4.** RDS : users table

id(primary)	email	title	category	contents	link
1	capstone@gmail.com	연구실 소개	대학원/학석연계	성균관대학교의 연구실을 소개합니다.	<a href="https://skku.edu">https://skku.edu</a>

**Table 5.** RDS : recommends table

데이터베이스에 등록된 사용자 관심도 정보를 기반으로 하루에 한번씩 모델의 학습이 이루어지며, 학습이 끝난 후에는 테이블에 기록된 정보가 초기화된다. 서버가 제공하는 세가지 기능을 정리하여 도식화한 그림은 ??를 참고하라.

### 4.3 Building Dataset for Machine Learning

Dataset을 구축하기 위한 설문조사 결과를 정리하면 다음 사진 ??와 같다. Dataset의 최종목표는 3가지 Local file을 구성하는 것이며 json 형식의 Feature 리스트인 Feature.json, csv 형식의 Target.csv, 그리고 Edge.csv로 나뉜다. Feature의 구성은 {id : [학년, 관심사]}의 순서로 되며 학년, 관심사의 Integer Mapping은 1학년에서 4학년이 각각 0에서 3에 매핑되고, ?? 내에 있는 카테고리에 학점관리에서 대학원까지의 카테고리가 각각 0에서 10까지 Integer에 매핑된다. 이후 총 1000개의 dummy node를

id(primary)	email	grade	interest	category_0	...	category_10
1	capstone@gmail.com	4	10	11	...	14

**Table 6.** RDS : visits table**Fig. 9.** model ui communication

만드는 것을 목표로 dataset을 만드는데, 만드는 과정은 다음과 같다. 1. 설문조사 결과에 dummy value 1씩 더한다. 2. 1, 2, 3 그리고 4학년에 해당하는 node들은 각각 250개씩을 목표로 만든다. 3. 각 250개 node들의 feature value 중 ‘관심사’ 항목은 설문조사 결과의 비율에 맞게 만든다. 4. 이렇게 11개의 feature로 나뉘어진 250개의 node들의 Target 값은 ‘선배추천’ 테이블의 비율에 맞춰서 다시 나눈다. 5. 이러한 과정에서 계산은 모두 Round Up으로 진행한다. 예시로 1학년의 ‘자기이해와 진로탐색’을 관심사로 가진 node의 개수는 250 개중  $(2+1)/(5+1)$  즉 약 18%로 46.875 즉 47개이며 이 47개중 target이 대학원/학석연계인 node의 개수는  $47 * (1+1)/(5+1) = 5.875$  즉 6개이다. 6. 마지막으로 features 가 같은 노드들끼리는 edge로 연결한다. Feature.json의 경우 : {"0": [0, 0], "1": [0, 0], "2": [0, 0], "3": [0, 0], "4": [0, 0], "5": [0, 0], "6": [0, 1], "7": [0, 1], "8": [0, 1], "9": [0, 1], ... } 1286": [3, 10], "1287": [2, 1], "1288": [4, 10], "1289": [1, 2]}로 이루어져 있으며 Edge.csv의 경우와 Target.csv 파일의 경우 아래의 Edge, Target 테이블과 같다. 이러한 방식으로 data Augmentation 했을 때 총 node의 개수는 1286개, edge의 개수는 56876개가 생성된다.

#### 4.4 Data Crawling and Provided Contents

데이터 크롤링은 Selenium과 BeautifulSoup4 를 활용했다. 총 11개의 카테고리중 10개의 카테고리에 대해서 성균관대학교 철린지스퀘어, 소프트웨어학과 홈페이지 등 19개 사이트의 23개 게시판에서 진행되었으며, 각 카테고리별 사이트와 정보의 개수는 다음 ??와 같다.

#### 4.5 ML model - N-GCN(Multi-scale Graph Convolution Network)

데이터셋으로부터 사용자의 정보를 입력받으면, 먼저 클릭한 횟수에 따라 기존 데이터베이스에 존재하는 해당 feature의 개수를 늘리는 형식으로 가중치를 부여한다. 그 후, 모델을 학습시키고, 추천 받아야 하는 사용자의 feature를 {"capstone@gmail.com":

선배추천	학년	1	2	3	4
카테고리	학점관리	12	16	7	4
	자기이해와 진로탐색	17	9	4	3
	봉사	3	2	2	0
	동아리, 학회	23	15	6	0
	외국어 점수 획득(국제품)	1	2	2	1
	자격증	1	0	0	2
	공모전/외부활동	6	10	11	8
	졸업요건	1	3	1	7
	기업인턴십/산학협력	0	2	10	8
	취업캠프/서류, 면접 클리닉	0	0	1	6
	대학원/학석연계	0	1	1	1
	sum	64	60	45	40
<hr/>					
후배관심사	학년	1	2	3	4
카테고리	학점관리	1	4	2	1
	자기이해와 진로탐색	2	2	3	5
	봉사	0	0	0	3
	동아리, 학회	1	5	3	1
	외국어 점수 획득(국제품)	0	0	0	0
	자격증	0	1	0	2
	공모전/외부활동	0	3	2	7
	졸업요건	0	0	2	6
	기업인턴십/산학협력	0	2	2	3
	취업캠프/서류, 면접 클리닉	0	1	1	6
	대학원/학석연계	1	0	0	1
	sum	5	18	15	35

Fig. 10. 설문조사결과

[2, 1]} 다음과 같은 형식으로 입력을 받아 {"capstone@gmail.com": [1, 2, 0]}의 형태로 출력을 한다. 그러면 해당 결과를 각 숫자에 맞는 카테고리 데이터 내에서 Content 5개를 뽑아 추천 데이터베이스에 전송을 해준다.

## 5 Evaluation

### 5.1 Evaluation metrics, Hyperparameters, and Input/Output for ML

Learning Rate = 0.01, epoch = 200, Training data = 750개, Validation data = 250개를 적용하여 모델을 학습시켰는데, overfitting이나 underfitting을 방지하기 위해

id_1	id_2
0	1
0	2
0	3
...	...
1	124
1	125
1	322
1	323
...	...
1286	996
1286	1098
1286	1207

**Table 7.** Edge.csv

id	target
0	0
1	0
2	0
3	0
...	...
1283	8
1284	9
1285	9
1286	10

**Table 8.** Target.csv

Early stopping을 사용했다. {"capstone@gmail.com": [2, 1]}의 input이 입력이 되면, {"capstone@gmail.com": [1, 2, 0]}의 형태로 추천 카테고리 top3를 출력해준다.

Evaluation Metric으로는 Accuracy하고 F1\_score를 사용했다. Accuracy = 0.875, F1 score = 0.868 정도로 나왔다. Accuracy를 사용한 이유는 결국 우리의 목적인 예측값과 결과값이 같아야하기 때문이다. 그리고 F1\_score를 사용한 이유는 아무리 Accuracy 값이 높게 나와도 F1\_score가 낮게 나오면 Accuracy 결과값은 의미가 없어지기 때문이다.

## 5.2 Empirical Result and Achievement for Project

여러 테스트 결과, 유저의 feature에 대해, 카테고리를 잘 추천해 주는 것을 볼 수 있었는데, 위의 예시에서 보듯이, '1' 카테고리에 대해 관심이 있는 사람에 대해 '1' 카테고리가 추천 1순위에 있는 것을 확인할 수 있다. 이것으로 개발 단계 중에 목표로 했던 카테고리를 추천해 그에 해당하는 콘텐츠를 뽑는 것에 성공을 했다고 볼 수 있다.

사이트, 제공 정보	카테고리	개수
레드커넥트 헌혈 어플리케이션	봉사	47
성균관대학교 교내봉사 ur1004		
성균관대학교 교내봉사프로그램 ur1004 공지사항		
1365 수원 봉사활동 리스트		
vms 수원 봉사활동리스트		
잡코리아 매출 1000대기업	취업캠프/서류,면접 클리닉	51
잡코리아 인턴 채용공고		
인크루트 신입 100대기업		
인크루트 인턴 채용공고		
사람인 정규 + 인턴 채용공고		
소프트웨어학과 졸업하고 뭐할까? 졸업생 노선		
성균관대학교 소프트웨어대학 취업, 인턴십 공고	기업인턴십/산학협력	18
툴게이트 인턴/산학협력 모집공고		
캠퍼스몬 공모전	공모전/외부활동	37
위비티 공모전		
대학원 연구소별 소식, 정보	대학원/학석연계	65
동아리 정보	동아리, 학회	98
토이, 오픽 안내	외국어 접수 획득(국제품)	11
학생성공 게이트웨이 해외유학생/교환학생 정보 및 상담 연계		
글로벌 하계학기 등 다양한 외국어 프로그램		
학생인재 개발팀 자기이해와 진로탐색을 위한 가이드북	자기이해와 진로탐색	8
챌린지 스퀘어 자기 이해와 진로탐색		
카운슬링센터	학점관리	65
교육개발센터 학습컨설팅		
챌린지스퀘어 비교과프로그램	졸업요건	5

**Table 9.** 데이터 크롤링 결과 요약

## 6 Limitation and Future works

프로젝트의 제한점과 가정사항들은 크게 Dataset 수집 및 제작, AI Model 구현, Frontend 개발, Backend 개발 분야로 나누어서 서술할 수 있다.

### 6.1 AI Model & Dataset

먼저, train dataset을 생성하는 과정에서 GNN 모델의 각 노드가 가지는 feature의 개수가 다양하지 않아, 각 user의 상태 정보를 자세하게 표현하지는 못했다. 또한 feature에서 사용자의 관심도를 확인하는 인자로서, 추천해줄 컨텐츠가 속해있는 카테고리를 사용하였기 때문에, 같은 카테고리 내에 속한 정보라면 사용자가 관심있게 여긴다는 가정을 내포하고 있다. 또한 카테고리 모델을 통해서 카테고리 범위까지의 추천을 완료한 후에는 해당 카테고리 내에서 랜덤한 컨텐츠를 추천해준다는 한계점도 존재한다. 위 한계점을 보완하기 위해서 앞으로 충분한 사용자 데이터가 축적된다면, feature를 더욱 다양화하고, GNN 모델의 노드를 컨텐츠 단위까지 확장시키는 방법으로 보다 정교한 추천을 할 수 있을 것이다.

### 6.2 Frontend

프론트엔드에서는 최근 많은 어플리케이션에서 사용하는 구글, 카카오, 네이버 등의 로그인 API를 사용하여 로그인하는 방식을 채택하지 못하였다. 해당 기업들에서 제공하는 로그인 API를 이용하면 처음 서비스를 접하는 유저들은 굳이 계정을 생성하고 비밀번호를 기억해야 하는 수고를 덜 수 있고 추후에도 어플리케이션에 로그인할 시

이메일과 비밀번호를 매번 적지 않고 몇 번의 클릭만으로 간단히 로그인할 수 있다. 본 프로젝트의 초기 목적중에는 사용자들에게 직접 사이트를 찾아가서 정보를 수집해야하는 수고를 덜어주기 위함도 있었으므로 당연히 어플 사용과정에서의 편의성은 중요한 요소라고 판단된다. 때문에 추후 서비스를 지속한다면 로그인 API를 이용한 로그인 기능을 추가할 예정이다.

### 6.3 Backend

마지막으로, 백엔드를 설계하면서 발생한 제한사항은 로그인 및 회원가입을 진행할 때 서버에 보내는 사용자 정보를 암호화하지 못한 것이다. 비밀번호와 사용자의 학사 정보 등은 외부에서 데이터를 가로챌 경우 보안적인 문제가 발생하므로, 암호화가 필요하지만 프로젝트를 수행하면서는 로그인과 회원가입의 기능만을 제공하도록 하였다. SSL 또는 TLS를 이용한 암호화 프로그램을 서버에 설치하거나, 해시 함수를 사용하여 UI로부터 비밀번호 등의 문자열을 받은 후에 다시 서버에서 해독하는 과정으로 로그인 및 회원가입 기능을 보완할 필요가 있다.

또한, 서버의 AI Model로부터 얻은 추천정보를 정시마다 사용자에게 전달하는 매 커니즘은 완전히 설계되었지만, 사람의 개입 없이 완전히 자동화된 방법으로 모델을 학습 및 작동시키려면 AI Model의 알고리즘을 서버상에 업로드하여야 한다. 초기에 이벤트가 발생할 때 동작을 수행하는 aws Lambda 서비스를 사용하여 딥러닝 모델을 서버상에 업로드하려고 했지만, 정시마다 추천정보를 업로드하기 위해서는 또다시 제 3자의 post 요청이 필요하였다. 따라서 백엔드에서의 또 하나의 Future work로서, aws EC2 서비스를 사용하여 서버상에 존재하는 컴퓨터에 추천 모델을 업로드하고, 이를 통해 완전히 자동화된 시스템을 구축할 수 있을 것이다.

## 7 Conclusion

지금까지 성균관대학교 학생들을 위한 진로활동 추천 애플리케이션인 Kingo Manager에 대해서 알아보았다. AWS RDS를 활용하여 사용자 정보, 추천정보, 학습 파라미터 데이터를 관리하였고, 주기적으로 데이터베이스에 접근해 학생들에게 알맞은 추천 정보를 업데이트해 줄 수 있도록 구현하였다. Kingo Manager가 다른 활동 추천 어플과 비교하였을 때 가장 큰 차이점은 N-GCN 모델을 활용해 추천을 해주는 점이다. 학습 과정은 어플리케이션에서 사용자의 컨텐츠 열람 횟수를 누적시킨 후에 데이터베이스에 업로드하면, 해당 활동 정보를 활용해 모델을 학습시켜 사용자에게 적합한 활동을 주기적으로 업데이트해주는 방식으로 설계하였다.

장기적으로, Kingo Manager은 성균관대학교의 대학생을 위한 진로활동 추천 서비스로서 사용자가 점점 증가할 수록 사용자 정보를 학습하는 과정을 통해 보다 정밀한 추천을 할 수 있다. 또한 서비스 지원 범위를 점점 확대해서 보다 일반적인 대학생들이 사용할 수 있는 진로활동 추천 서비스로 자리잡을 것이다.

## List of Figures

1	frontend architecture .....	4
2	backend architecture .....	5
3	Model .....	8
4	SplashScreen .....	9
5	Login page .....	9
6	Signup pages .....	11
7	Main pages.....	12
8	Settings pages .....	13
9	model ui communication .....	15
10	설문조사결과 .....	16

## List of Tables

1	Requirements .....	3
2	Survey .....	7
3	Async-storage .....	14
4	RDS : users table .....	14
5	RDS : recommends table .....	14
6	RDS : visits table .....	15
7	Edge.csv .....	17
8	Target.csv .....	17
9	데이터 크롤링 결과 요약 .....	18

## References

- [1] Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019, January 24). *Session-Based Recommendation with Graph Neural Networks*. Retrieved December 4, 2022, from <https://arxiv.org/abs/1811.00855>
- [2] Sami, A., Nazanin, A., Hrayr, H., Amol, K., Bryan P.: A *Higher-Order Graph Convolutional Layer*, In: University of Southern California, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montreal, Canada.
- [3] N-GCN Homepage, <https://github.com/benedekrozemberczki/MixHop-and-N-GCN>.