캡스톤디자인(종합설계) 결과보고서

소속학부(과)	디지털콘텐츠공학과	팀명	www
개설 연도 및 학기	2022 학년도 ■1학기 □2학기	교과목명	캡스톤디자인2
과제명	OTT 컨텐츠 소비자 맞춤 리뷰 및 홍보 AI 서비스		
과제유형	■기업연계형 캡스톤디자인	□기술이전형 캡스톤디자인	
희망금액		(기술이전금액)천원	

참여기업현황					
기업	기업명		소재지	
	사업자번호		주요생산품목	
담당자	성명		소속부서	
	H.P		E-mail	

참여 학생 현황

구분	이름	학부(과)	학년	학번	H.P	E-mail
팀장		디지털콘텐츠공학과	4			
팀원1		디지털콘텐츠공학과	4			
팀원2		디지털콘텐츠공학과	4			
팀원3		디지털콘텐츠공학과	4			
팀원4						
팀원5						
팀원6						
팀원7						

집행경비내역	비목	집행내역	금액
	재료비		천원
	인쇄비	*문헌구입비*	100천원
	학생여비	자세히 작성	
	학생회의비	( )천원 × ( )인 × ( )회	천원
			천원
	총액		천원

위와 같이 캡스톤디자인(종합설계) 결과보고서를 제출합니다.

첨부 : 캡스톤디자인(종합설계) 과제 상세 결과보고서[별첨 1호]

20 년 월 일

지원학생(팀장) (인)

사업책임자(지도교수) (인)

참여기업 담당자 (인)

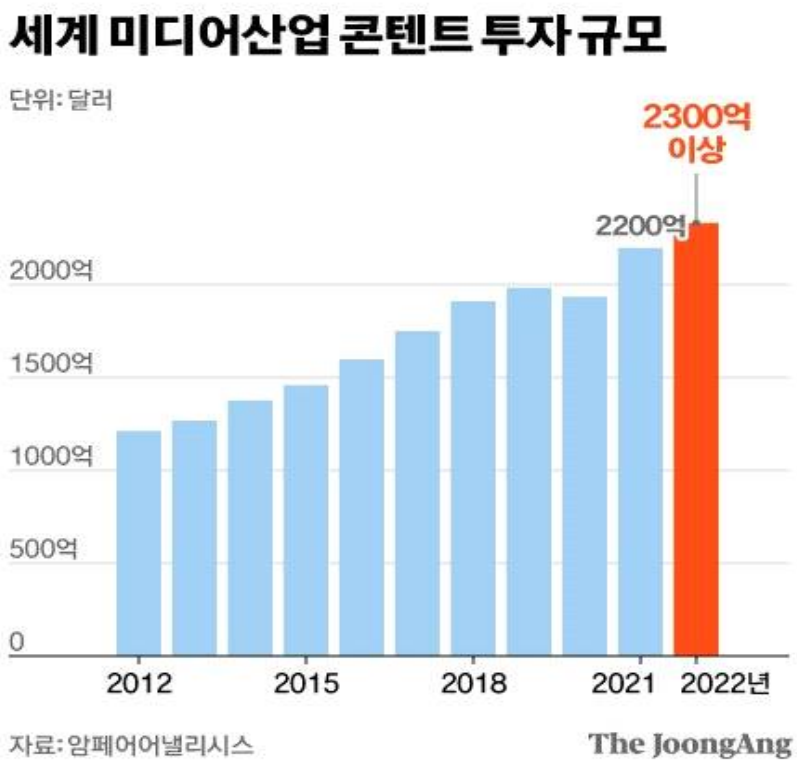
원광대학교 LINC 3.0 사업단장 귀하

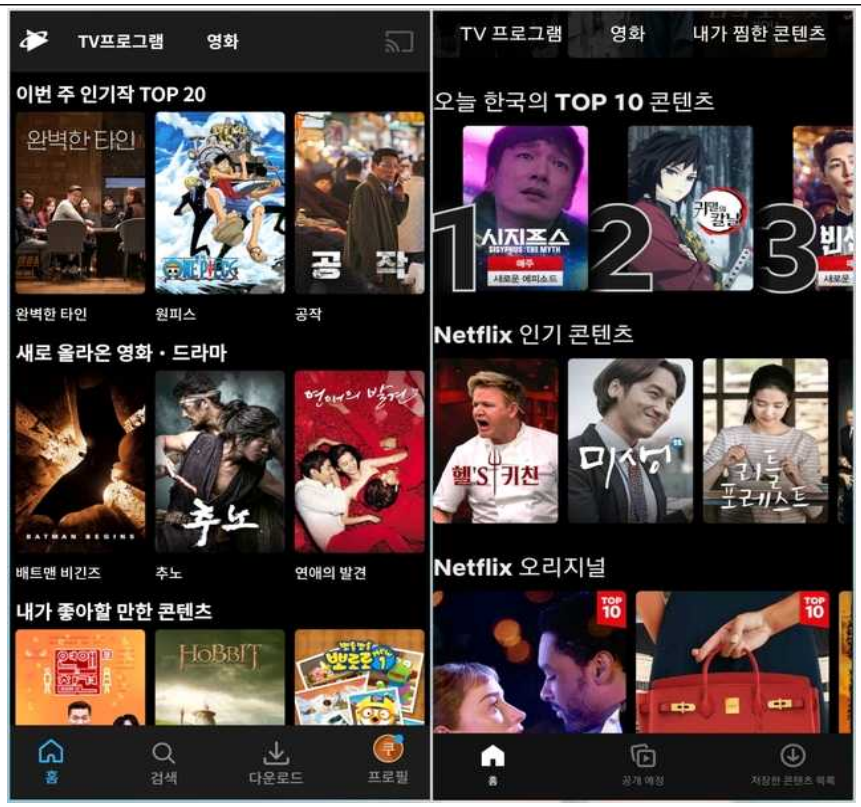
- 기업연계프로젝트 수행 주제의 선정 이유

OTT(Over The Top)서비스는 인터넷을 통해 언제 어디서나 방송/프로그램 등의 미디어 컨텐츠를 시청(소비)할 수 있는 사용자 중심적인 서비스이다.

OTT 등장의 배경에는 시청자들, 콘텐츠 시장의 소비자들이 현재의 방송 서비스로 채우지 못하는 수요 그 자체가 있다. 만약 소비자들이 현재의 서비스에 충분히 만족하고 있다면 새로운 방송 서비스가 들어설 자리는 없을 것이다. 하지만 현재의 모든 방송 서비스는 한 가지 한계를 갖고 있다. 시청자는 어떤 방송사의 특정 시간대에 선택 가능한 어떤 채널을 선택할 뿐, 엄밀히 말해 컨텐츠 자체를 선택하는 것은 아니다. 보고 싶은 영화가 있어도 그것을 보여 주는 채널이 없다면, 또는 그 상영 시간이 이미 지나갔다면 그 영화를 볼 수 없다.

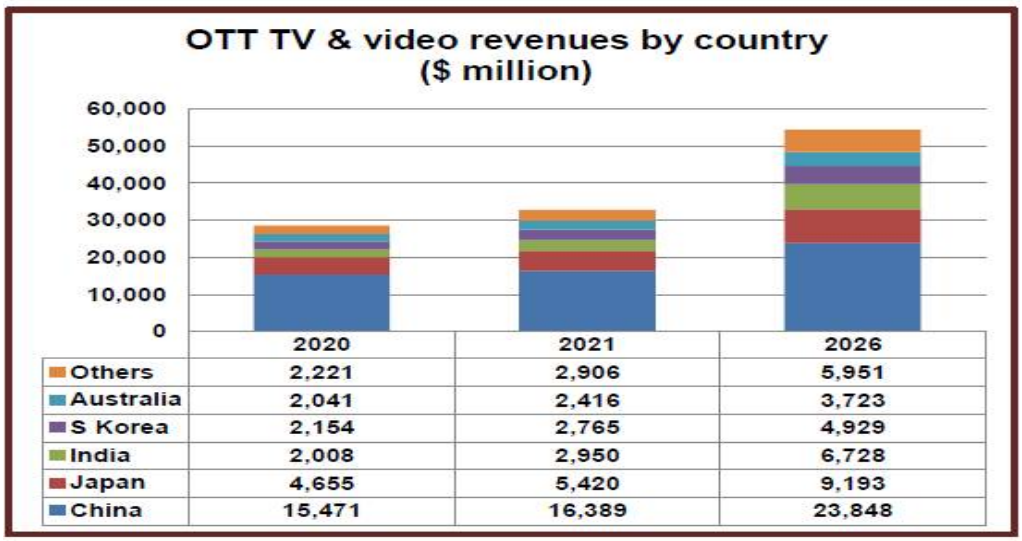
최근 **초고속 인터넷 네트워크 서비스의 발전과 스마트 디바이스의 진화**로 동영상 시청에 불편함이 사라지면서 OTT 시장이 성장했다. 국외 기업 넷플릭스, 디즈니플러스와 국내 기업 TVING, Wavve, 쿠팡플레이 등의 시장 가치가 증가했다.





▲ 쿠팡플레이, 넷플릭스 홈 화면

하지만 OTT서비스에도 한 가지 문제점은 존재한다. 요즘날 OTT 플랫폼들은 볼거리가 넘쳐남에도 불구하고 이용자들은 여전히 볼 것이 없다고 말한다. 넘쳐나는 콘텐츠 사이에서 '풍요 속 빈곤' 현상을 마주한 것이다. 예를 들어 최근 소셜미디어(SNS)상에서는 넷플릭스는 "볼건 많은데 볼건 없네" 왓챠는 "이 영화가 없다고?, 이 영화가 있다고?" 웨이브는 "아, tvn없네" 등 OTT 플랫폼 사용 후기에 대한 반응이 공감을 얻고 있다. 이러한 현상은 자신의 취향에 맞는 영상을 찾지 못했기 때문에 발생한다.



	2020	2021	2026
Others	2,221	2,906	5,951
Australia	2,041	2,416	3,723
S Korea	2,154	2,765	4,929
India	2,008	2,950	6,728
Japan	4,655	5,420	9,193
China	15,471	16,389	23,848

아시아태평양의 OTT TV 및 비디오 시장 예측
출처 : Digital TV Research

콘텐츠 기반 추천 시스템은 1997년 무비렌즈를 통해 처음 적용된 알고리즘으로, 2006년 넷플릭스의 추천 알고리즘 경진대회 등을 통해 대중들에게 AI 추천 알고리즘이 본격적으로 등장했다. 대표주

자인 유튜브, 넷플릭스의 시청자 선택 비율은 76-90프로 내외로 매우 높았다. 구글의 유튜브는 추천 후보생성 모델과 순위평가 네트워크의 두 개의 신경망으로 구성되어있다. 추천 후보 생성 모델은 사용자 파악 개별 정보인 사용자의 동영상 ID, 검색 키원, 사용자의 위치, 나이, 성별과 시청 이력을 파악하여 선호하는 콘텐츠 시청 트렌드를 파악하여 심층 신경망에 입력된다. 추천모델 알고리즘은 시청자가 선호하는 예측 범위 내에 존재하는 비디오에서 시청할 가능성이 높은 비디오를 선정한 후 확률이 높은 순으로 추천한다.

순위평가 모델은 쿠팡, 스포티파이, 우버이츠, 호텔닷컴 등의 플랫폼에서 마케팅 도구로 활용하는 시스템과 유사하다. 추천한 영상 중에 시청되지 않은 영상은 배제되어 다음 추천 리스트를 표시하는 프로세스가 반복되며 동영상을 맞춤식으로 제공한다.

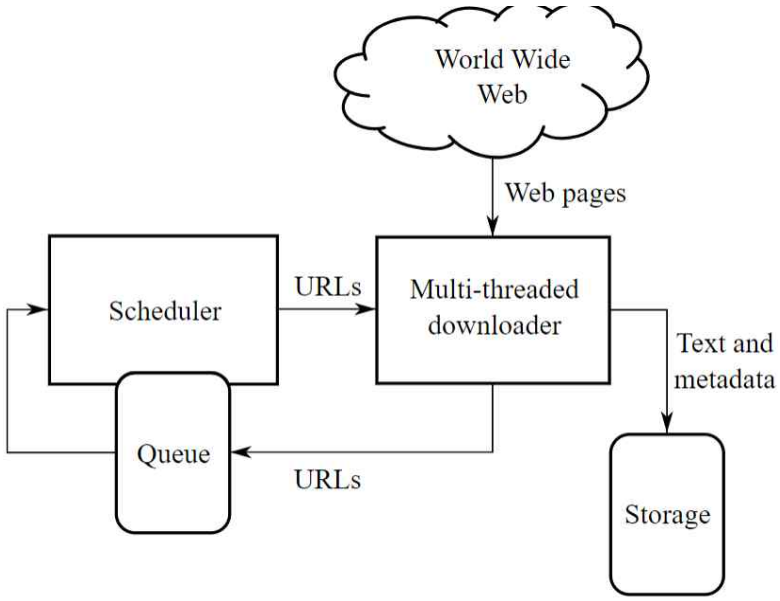
4, 연구 목표

5. 관련 연구 분석

- 본 과제와 관련된 기존의 연구 내용, 관련 특허, 관련 제품에 대한 자료 조사 내용 기술

6. 연구 수행 내용

우리가 만들 'OTT 콘텐츠 소비자 맞춤 리뷰 및 홍보 AI 서비스'는 크롤링 기술과 Node, Vue, MySQL을 사용하여 구현하고자 한다.



▲ WEB crawling

크롤링(crawling) 혹은 **스크레이핑(scraping)은** 웹 페이지를 그대로 가져와서 거기서 데이터를 추출해 내는 행위다. 크롤링하는 소프트웨어는 크롤러(crawler)라고 부른다.

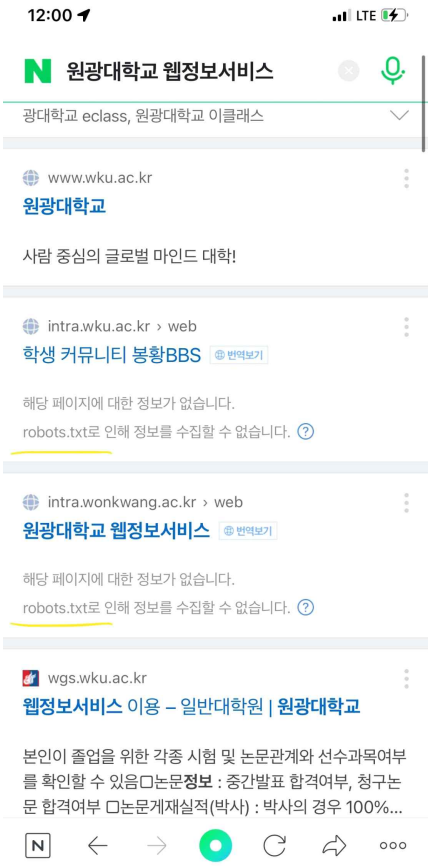
검색 엔진에서도 유사한 것을 필수적으로 사용하는데, 웹 상의 다양한 정보를 자동으로 검색하고 색인하기 위해 사용한다. 이때는 스파이더(spider), 봇(bot), 지능 에이전트라고도 한다. 사람들이 일일이 해당

사이트의 정보를 검색하는 것이 아니라 컴퓨터 프로그램의 미리 입력된 방식에 따라 끊임없이 새로운 웹 페이지를 찾아 종합하고, 찾은 결과를 이용해 또 새로운 정보를 찾아 색인을 추가하는 작업을 반복 수행한다. 방대한 자료를 검색하는 특징이 있고, 네이버, 구글 등도 이런 봇을 이용해 운영된다.

<크롤링 정보로 앱 개발 시의 저작권 문제>

크롤링은 합법적 크롤링과 불법적 크롤링으로 구분된다. 이 기준은 사이트 운영자의 의사에 반하는지에 따라 구분될 수 있다. 사이트의 운영자의 동의를 받았다면 합법적 크롤링이고, 동의를 받지 않았다면 불법적인 크롤링이라 볼 수 있다.

특히 웹사이트의 운영자가 웹서버의 홈 디렉터리에 위치한 robots.txt(웹사이트 검색에는 로봇이 접근하는 것을 방지하는 규약으로 robot.txt코드를 넣어 외부에서 데이터를 읽고 접근하는 것을 막는 조치) 파일에 포괄적인 크롤링 금지 또는 특정 검색엔진의 크롤링 금지, 특정 디렉터리에 대한 크롤링 금지를 표시하였음에도 불구하고, 그 표시를 무시하고 크롤링을 할 경우에는 불법적인 크롤링에 해당한다. 불법크롤링은 데이터베이스권 침해에도 해당할 수 있다.



1. 웹사이트를 무단으로 크롤링하여 게시하는 것은 저작권 침해가 성립될 수 있고 이는 해외에 있는 웹사이트도 마찬가지이다. 따라서 크롤링하는 웹사이트의 허락이 필요하다.
2. 출처를 남긴다고 하더라도 해당 웹사이트를 크롤링하여 게시하는 것을 상업적으로 활용하는 것은 저작권 침해가 성립될 수 있고 따라서 동의가 필요하다.
3. 해외에 있는 회사가 저작권 침해를 이유로 민사 또는 형사소송을 진행할 수 있다. 따라서 원작자의 동의를 받으시는 것이 필요하다.
4. 웹크롤링을 통해 수집해오는 것 자체가 저작물에 해당하지 않더라도 웹사이트가 데이터베이스에 해당할 소지는 있다는 법원(서울고등법원 2016나2019365판결)의 판단이 있었다.

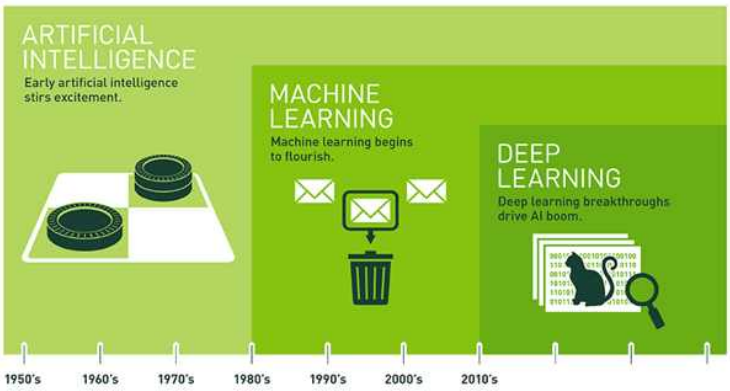
5. 다만 교육, 학술 또는 연구, 시사보도를 위하여 이용하는 경우(영리목적은 제외)등에는 데이터베이스제작자의 권리를 침해한 것으로 인정되지 않는다.

실제 한 법원 판례를 보면, 무단으로 크롤링 한 정보에 대해 출처를 명시함에도 불구하고, 출처 웹사이트로 방문하지 않아도 검색이 가능한 점과, 데이터베이스 양과 질,방문자 수나 이용시간이 중요한 영향을 미치는 점을 종합하면 데이터 베이스 제작자의 권리 침해가 있다고 판시하였다.

<웹크롤링 저작권 문제 해결 방안>

1. 웹사이트 운영자, 원작자의 동의를 구한다.
2. 출처사이트를 방문해야만 하는 구조로 제작한다.
3. OTT 콘텐츠 리뷰 플랫폼에 리뷰어들이 게시를 직접한다.

리뷰 서비스 추천 AI



인공지능 : 기계를 지능적으로 만드는 과학기술, 기계는 문제를 해결할 때 알고리즘을 기반으로 문제를 해결하게 됨. AI 알고리즘은 규칙이 생성되는 방식에서 기존 알고리즘과 차이가 있음.

기존 알고리즘은 입력값들에 대한 출력을 정의하는 특정 규칙을 설정하는 반면 AI 알고리즘은 자체 규칙 시스템을 구축하게 됨.

머신러닝 : 정확한 결정을 내리기 위해 **제공된 데이터를 통하여 스스로 학습할 수 있음**. 처리될 정보에 대해 더 많이 배울 수 있도록 많은 양의 데이터를 제공해야함.

즉, **빅데이터를 통한 학습 방법**으로 머신러닝을 이용할 수 있음. 머신러닝은 기본적으로 알고리즘을 이용해 데이터를 분석하고, 분석을 통해 학습, 학습한 내용을 기반으로 판단이나 예측을 함. 궁극적으로 의사 결정 기준에 대한 구체적인 지침을 직접 코딩해 넣는 것이 아닌, 데이터와 알고리즘을 통해 컴퓨터 그 자체를 '**학습**'시켜 작업 수행 방법을 익히는 것을 목표로 함.

딥러닝 : 인공신경망에서 발전한 형태의 인공 지능으로, 뇌의 뉴런과 유사한 정보 입출력 계층을 활용해 데이터를 학습함. 그러나 기본적인 신경망조차 굉장한 양의 연산을 필요로 하는 탓에 딥러닝의 상용화는 초기부터 난관에 부딪힘.

1단계 데이터 수집

- 기계에 학습시키기 위한 데이터를 준비하는 과정

- 수집된 데이터들은 후에 전처리 단계에서 정제 과정을 거치기 때문에 수집할 데이터는 최대한 많고 다양할수록 좋음.

2단계 데이터 탐색

- 실제로 해당 데이터에 대해 분석하는 과정.
- 데이터 간의 서로 가지는 관계, 필요 없는 데이터는 무엇인지 등으로 데이터 자체를 이해하는 과정으로 EDA(Exploratory Data Analysis)라고 불림.

3단계 데이터 전처리

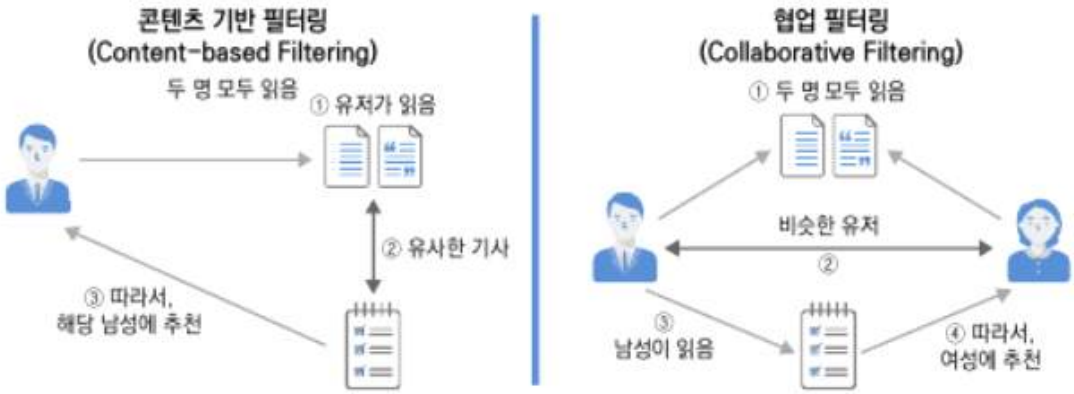
- 데이터 탐색 과정이 끝나면, 전처리 과정에 들어감
- 해당 데이터 중 학습에 악영향을 끼칠 만한 데이터를 제거하거나, 반대로 머신러닝 판단에 기여할 만한 눈에 띄는 특성들을 부각시키는 작업도 모두 포함될 수 있음.

4단계 데이터 학습

- 전처리를 포함한 데이터에 대한 모든 준비 과정이 완료되었다면, 실제 모델에서 데이터를 학습시킴.
- 적절한 기존 머신 러닝 알고리즘을 선정할 수 있고, 또는 자신이 직접 알고리즘을 설계할 수도 있음.
- 이 과정에서 해당 모델이 학습을 수행할 때, 사용자 측에서 결정해 주어야 하는 하이퍼파라미터 (Hyperparameter) 등을 결정함.
- 학습이 끝난 모델은 지금까지 훈련했던 데이터를 바탕으로 새로운 데이터에 대하여 예측을 하는 등의 결과를 보여줄 수 있게 됨.

5단계 모델 검증

- 학습된 모델은 예측한 결과를 제시해주지만, 이것이 잘 학습된 것인지 아닌지를 판단하기 위해서는 만들어진 모델에 대하여 평가를 진행해야 함.
- 평가 시에 사용되는 데이터는 일종의 “문제”이며, 이 문제는 이전학습 과정에서는 공개하지 않았던 데이터(Test Set)여야만 올바른 평가 진행이 가능함.
- 만약 평가 후, 기존에 계획했던 성능을 충족시키기 못할 경우, 전처리 내지 학습과정으로 다시 돌아가 다시 설계를 하고 학습을 진행한다.



▲ 콘텐츠 기반 필터링과 협업 필터링 (출처: Software carpentry)

AI 챗봇

챗봇은 채팅(chatting)과 로봇(robot)의 합성어로, 인공지능이 관여하여 상호 대화하는 플랫폼 서비스를 말한다.

인공지능은 금융 서비스 사기 탐지, 소매업의 구매 예측, 온라인 고객 지원 상호 작용과 같은 일상에서 응용되고 있다. 그 중 웹 페이지에서 챗봇을 통해 대화를 시작하는 경우 전문 AI를 실행하는 컴퓨터와 상호 작용하는 경우가 많다.

[표 1] 주요 챗봇의 개발현황

플랫폼	챗봇 개발 현황
카카오톡(Kakaotalk)	• 기존에 기업의 홍보와 관련된 자동응답 API를 사용하고 있음 • 24시간 금융상담 지원가능한 인터넷전문은행 '카카오 뱅크'와 연동예정인 '금융봇' 공개 예정
텐센트 위챗(WeChat)	• 텐센트의 메신저 프로그램, 실시간 주요도로 교통정보 제공 • 음식주문, 호텔, 항공예약 등의 O2O(Online to Offline)서비스 제공중
MS 샤오아이스(Xiaoice)	• MS에서 개발한 챗봇, 실제 사람과의 대화처럼 매끄러운 대화를 할 수 있도록 한 빅데이터 기반의 앱
킥 봇샵(Bot Shop)	• 화장품, 의류업체들과 연계한 쇼핑 앱 • 각각의 회사를 태그하면 회사에서 상품에 대한 질문과 주문을 처리하는 서비스 제공중
페이스북 와츠앱(WhatsApp)	• 전자상거래 업체 '소피파이', CNN과 제휴하여 쇼핑과 뉴스를 통틀어서 이용할 수 있는 서비스 제공 예정
라인(Line)	• LG와의 협업을 통해 IoT 서비스'홈챗'을 출시, 인공지능 봇을 이용한 스마트폰 콜센터 구축 예정 • 챗 인공지능 플러그인을 통해 여러 홍보용 자료를 배포할 수 있는 기반 마련
텔레그램(Telegram)	• Bot API 공개로 개발자들에게 챗봇 개발 지원 • 대화창에서 바로 이용이 가능한 Inline Bots를 추가
구글 알로(Allo)	• 인공지능 챗봇 기술이 적용된 메신저 플랫폼 준비중

* 출처 : 인공지능 기반의 '챗봇(ChatBot)' 서비스 등장과 발전 동향, 한국정보화진흥원, 2016. 8.

▲주요 챗봇의 개발 현황

챗봇 전문 에이전트(ex) Microsoft bot framework) 뿐 만 아니라 카카오톡, 라인, 네이버톡톡, 페이스북 등의 다양한 메신저 사업자들이 직접 챗봇을 만들 수 있는 툴을 제공하기도 한다.

[표 2] 텍스트기반 챗봇 주요 핵심 기술

관련기술	주요내용
패턴 인식(Pattern Recognition)	기계에 의하여 도형, 문자, 음성 등을 식별시키는 것.
자연어처리 (Natural Language Processing)	인간이 보통 쓰는 언어를 컴퓨터에 인식시켜 처리하는 일. 정보검색, 질의응답, 시스템 자동번역, 통역 등이 포함.
시멘틱 웹(Symantic Web)	컴퓨터가 정보자원의 뜻을 이해하고, 논리적 추론까지 할 수 있는 차세대 지능형 웹
텍스트 마이닝(Text Mining)	비정형 텍스트 데이터에서 새롭고 유용한 정보를 찾아내는 과정 또는 기술
상황인식컴퓨터(Text Aware Computing)	가상공간에서 현실의 상황을 정보화하고, 이를 활용하여 사용자 중심의 지능화된 서비스를 제공하는 기술

* 출처 : '모바일시대를 넘어 AI시대로', 한국정보화진흥원, 2010. 8.

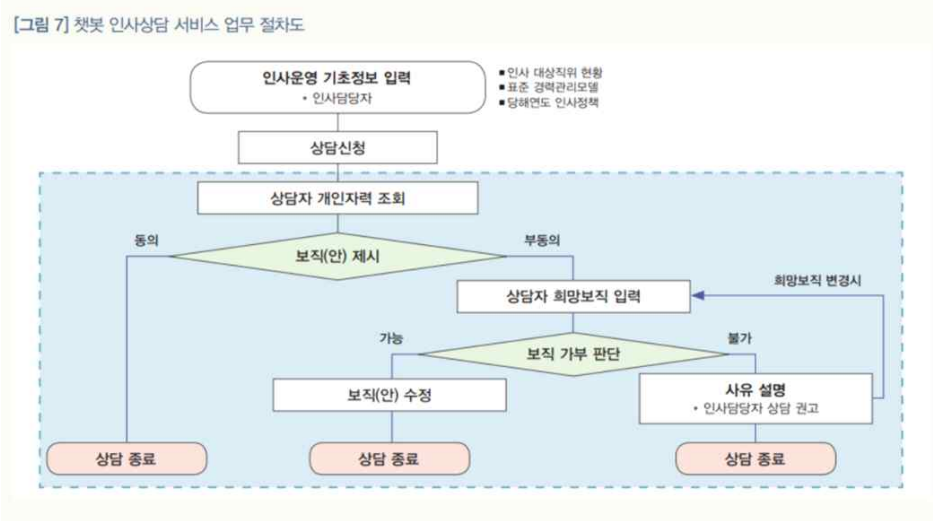
[표 3] 챗봇의 5대 활용 분야

분 야	종 류	관련 기업
대화형 커머스 및 O2O	쇼핑, 비행기예약, 숙소예약, 레스토랑 예약 및 주문, 택시 호출 등	Amazon, eBay, FB, 카카오, 인터파크
개인비서 서비스	헬스케어, 뉴스피드, 날씨정보, 금융상담, 일정관리, 길찾기 등	Google, MS, Pancho, CNN, 샤오빙, Skype
공공 서비스	법률상담, 세금납부, 부동산정보, 구인구직	법무부, 경기도 정보기획실, 미야(Mya),
엔터테인먼트 서비스	광고, 미디어, 방송안내, 데이팅, 공연 등	WeChat
기업용 메신저	정보검색, 파일공유, 데이터보관, 팀원정보 공유, 자동 사무화(OA), CRM	Slack, CareerLark,Growbot, Wework

▲챗봇 주요 핵심 기술과 활용 분야

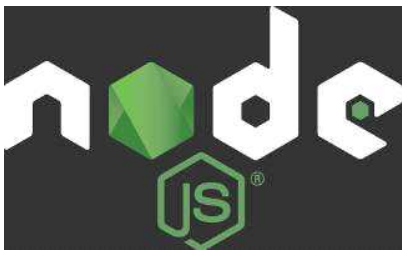
위 표와 같이 패턴인식, 자연어처리 시멘틱 웹 등 기술의 집합체인 챗봇은 사용성이 매우 높고

여러 분야에서 사용될 수 있다. 이를 토대로 금융, 쇼핑, 의료, 교통, 배달, 민원 등의 다양한 산업 분야에 대해 챗봇이 상담을 대신 처리하는 것이 가능하다. 또한 고객이 원하는 정확한 정보 전달과 함께 주문, 결제, 진료, 상담 등의 다양한 역할 수행이 가능하다. 실제 국내 많은 쇼핑몰에도 챗봇이 적용되어 상담율을 높이고 구매율을 증가시켰다.



▲챗봇 인사상담 서비스 업무 절차도

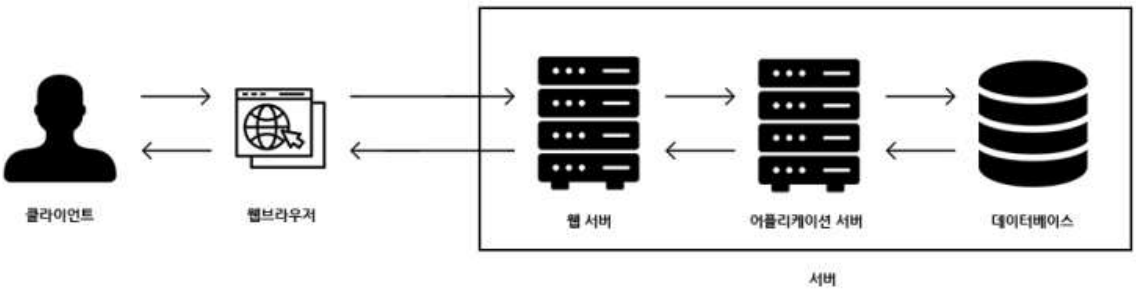
위와 같은 절차로 챗봇의 서비스 업무가 진행된다. 챗봇이 해석하거나 해결할 수 없는 문제 는 사람(직원)이 개입하여 사람과 직접 의사소통하게 된다. 이와 같이 해석되지 않은 사례는 머신 러닝 컴퓨팅 시스템에 입력되어 향후 상호 작용에 대한 AI 애플리케이션을 개선한다.



▲ Node.js

▲ Vue.js

▲ MySQL



▲ 웹 서버 구성도

Node.js는 확장성 있는 네트워크 애플리케이션 개발에 사용되는 소프트웨어 플랫폼이다. 작성 언어로 자바스크립트를 활용하며 논블로킹 I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있다.

Vue.js는 웹 애플리케이션의 사용자 인터페이스를 만들기 위해 사용하는 오픈 소스 프로그레시브 자바스크립트 프레임워크이다. Vue.js는 고성능의 싱글 페이지 애플리케이션(SPA)을 구축하는데 이용가능하다.

MySQL은 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템이다. 다중 스레드, 다중 사용자 형식의 구조질의어 형식의 데이터베이스 관리 시스템으로서 오라클이 관리 및 지원하고 있으며, Qt처럼 이중 라이선스가 적용된다.

Node.js로 서버를 생성하고 MySQL을 통해 데이터베이스를 생성,관리하여 고객 데이터를 수집한다. 수집한 데이터는 고객 추천 알고리즘과 챗봇 AI에 활용하는 등 서비스를 확대한다.

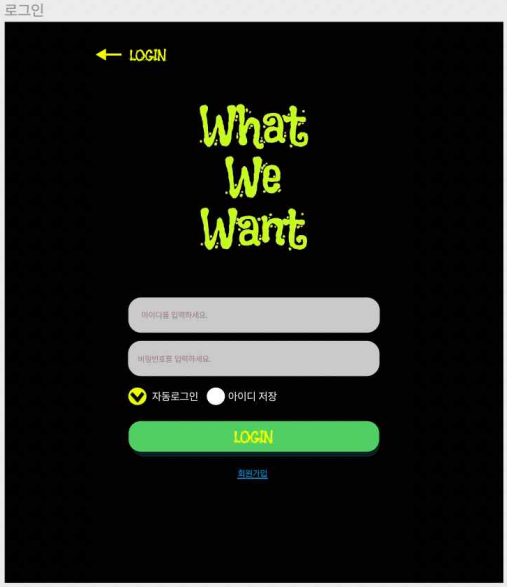
개발환경

프론트엔드	Vue.js
백엔드(서버)	Node.js
데이터 베이스(DB)	MySQL
IDE	VisualStudio Code
디자인	Figma

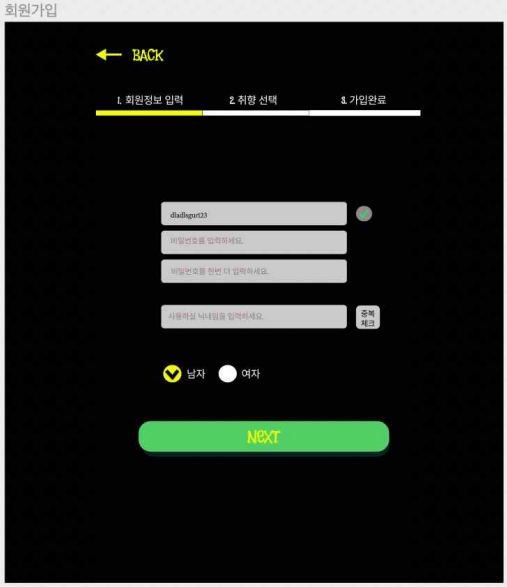
Node.js 설치 : https://nodejs.org/ko/download/
VisualStudio Code 설치 : https://code.visualstudio.com/download
Vue 설치 : VisualStudio Code 터미널 npm install vue 입력
VueCLI 설치 : VisualStudio Code 터미널에서 npm install @vue/cli-init -g 입력
로컬호스트 서버 실행 : 생성된 VueCLI 프로젝트 파일로 이동 후 npm run serve 입력 후 크롬에 localhost:8080을 url 입력

5주차 : 웹 디자인 : Figma

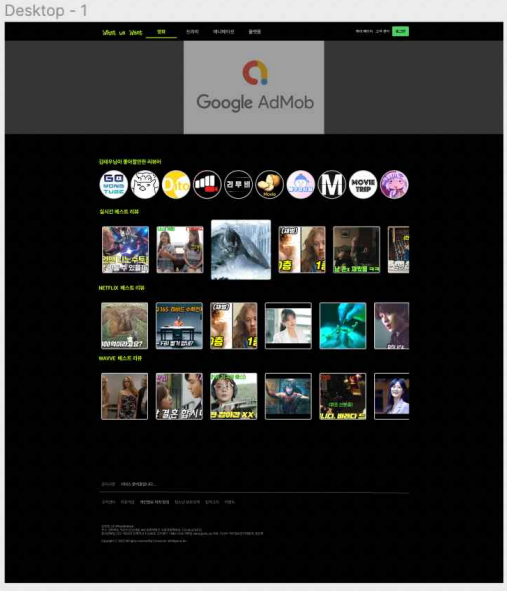
로그인



회원가입



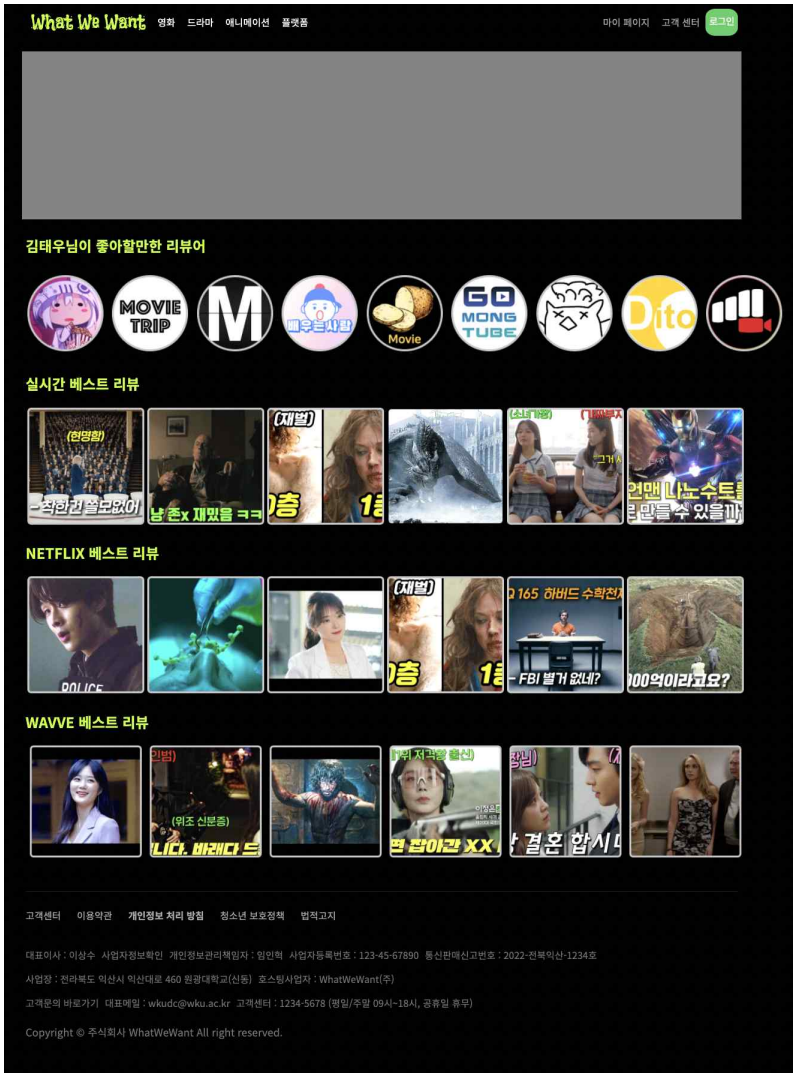
Desktop - 1



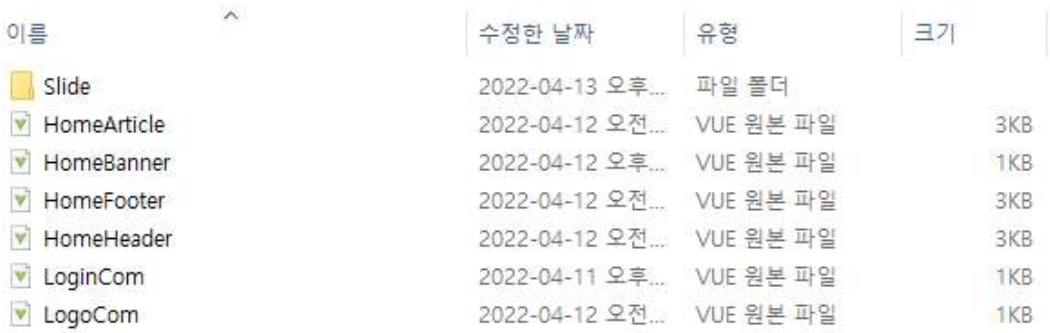
선택

6주차 : 메인페이지 개발

결과물



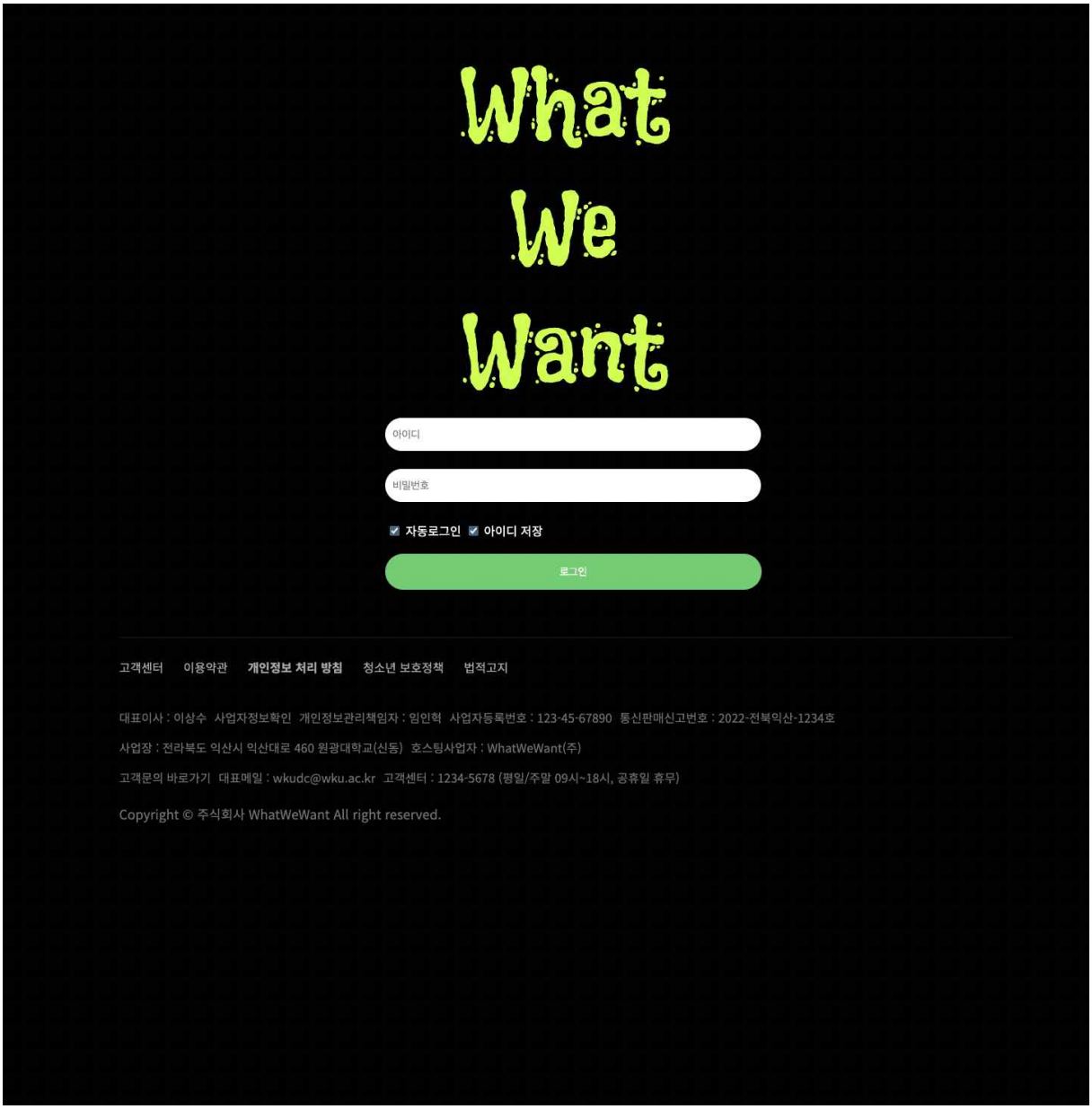
컴포넌트 구조



이름	수정한 날짜	유형	크기
BestSilde	2022-04-12 오후...	VUE 원본 파일	1KB
NetflixSlide	2022-04-12 오후...	VUE 원본 파일	1KB
RecomSlide	2022-04-12 오후...	VUE 원본 파일	2KB
WavveSlide	2022-04-12 오후...	VUE 원본 파일	2KB

7주차 : 로그인 창

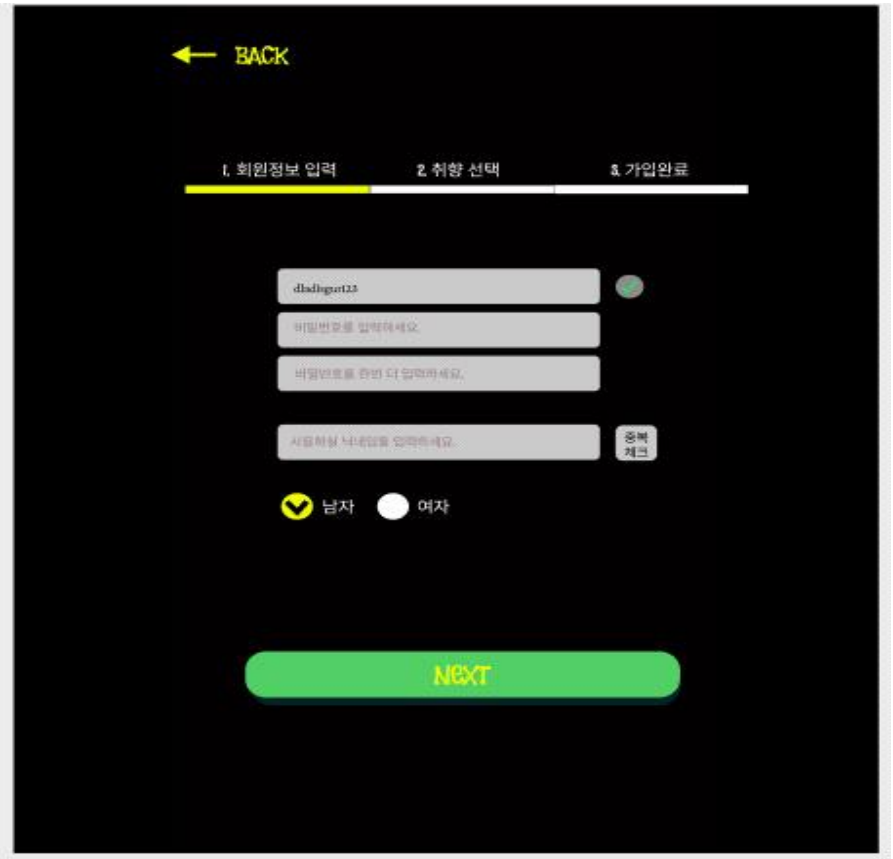
결과물

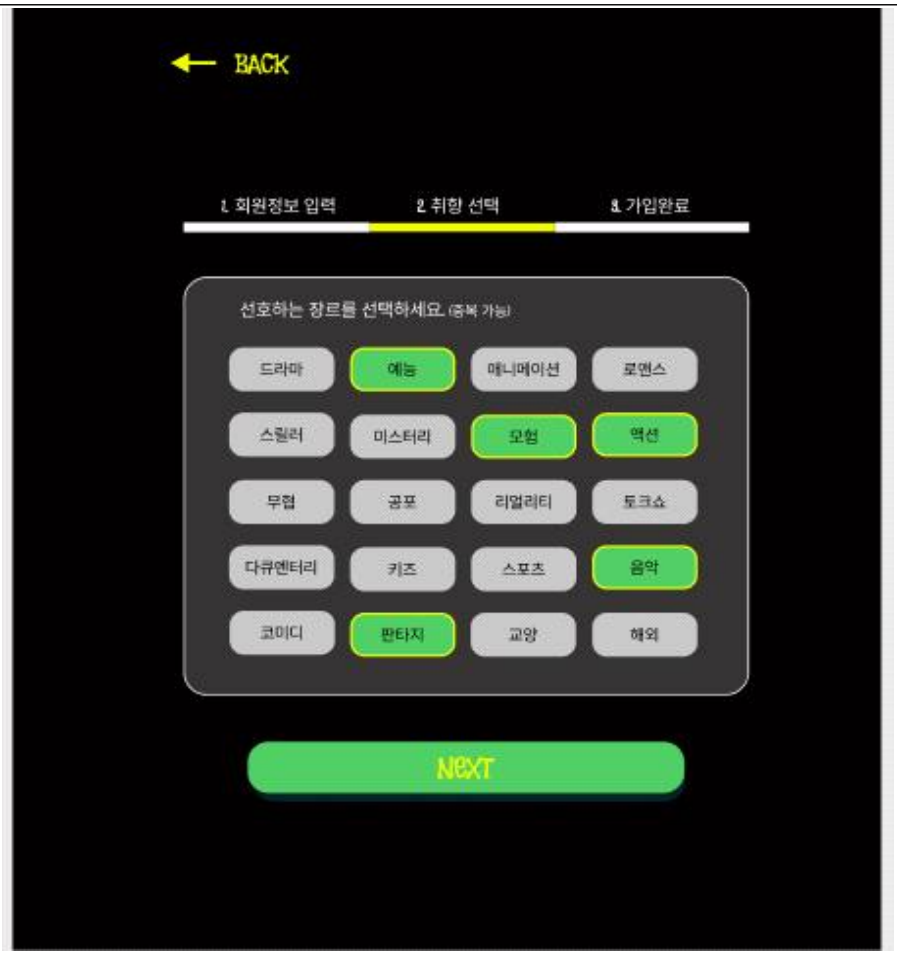


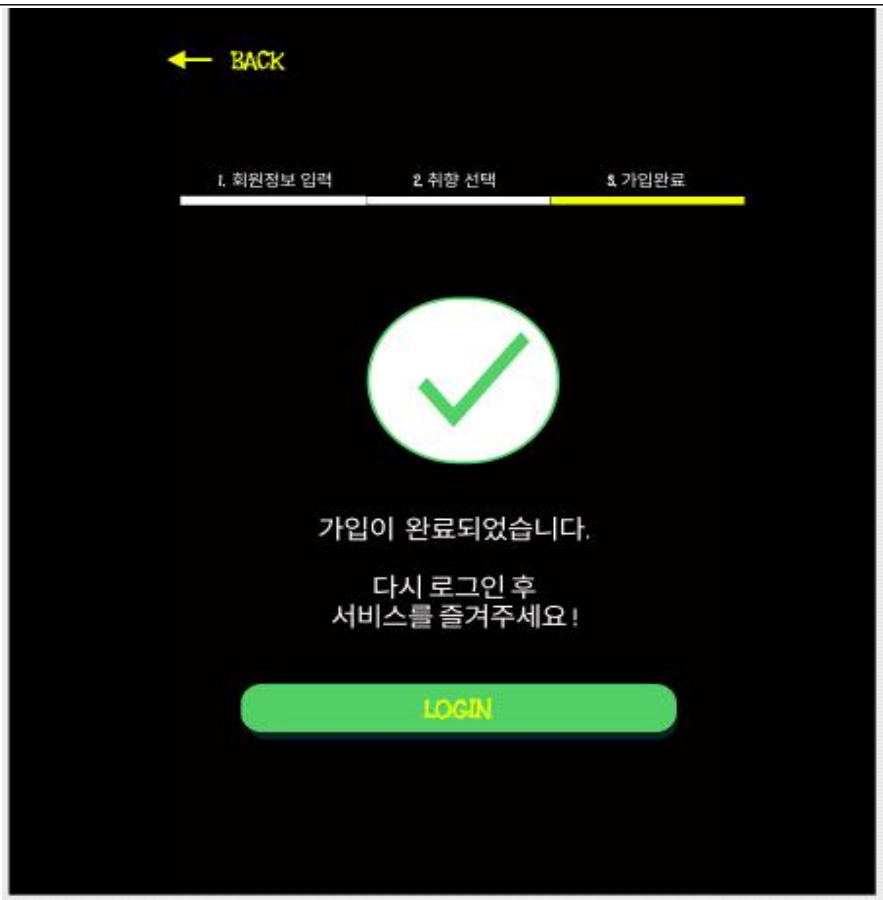
컴포넌트 구조

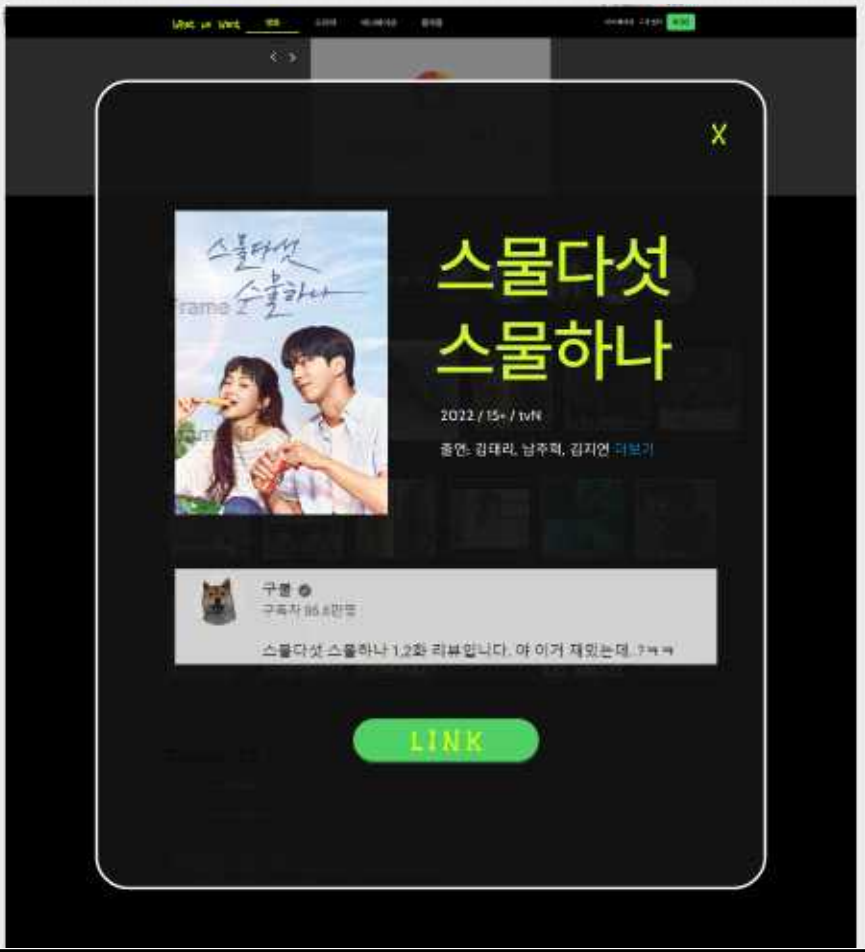
이름	수정한 날짜	유형	크기
LoginArticle	2022-04-12 오후...	VUE 원본 파일	3KB

9주차 : 회원가입 양식, 콘텐츠 정보창




BACK
1. 회원정보 입력
2. 취향 선택
3. 가입완료
선호하는 장르를 선택하세요. (중복 가능)
드라마
예능
애니메이션
로맨스
스릴러
미스터리
모험
액션
무협
공포
리얼리티
토크쇼
다큐멘터리
키즈
스포츠
음악
코미디
판타지
교양
해외
NEXT

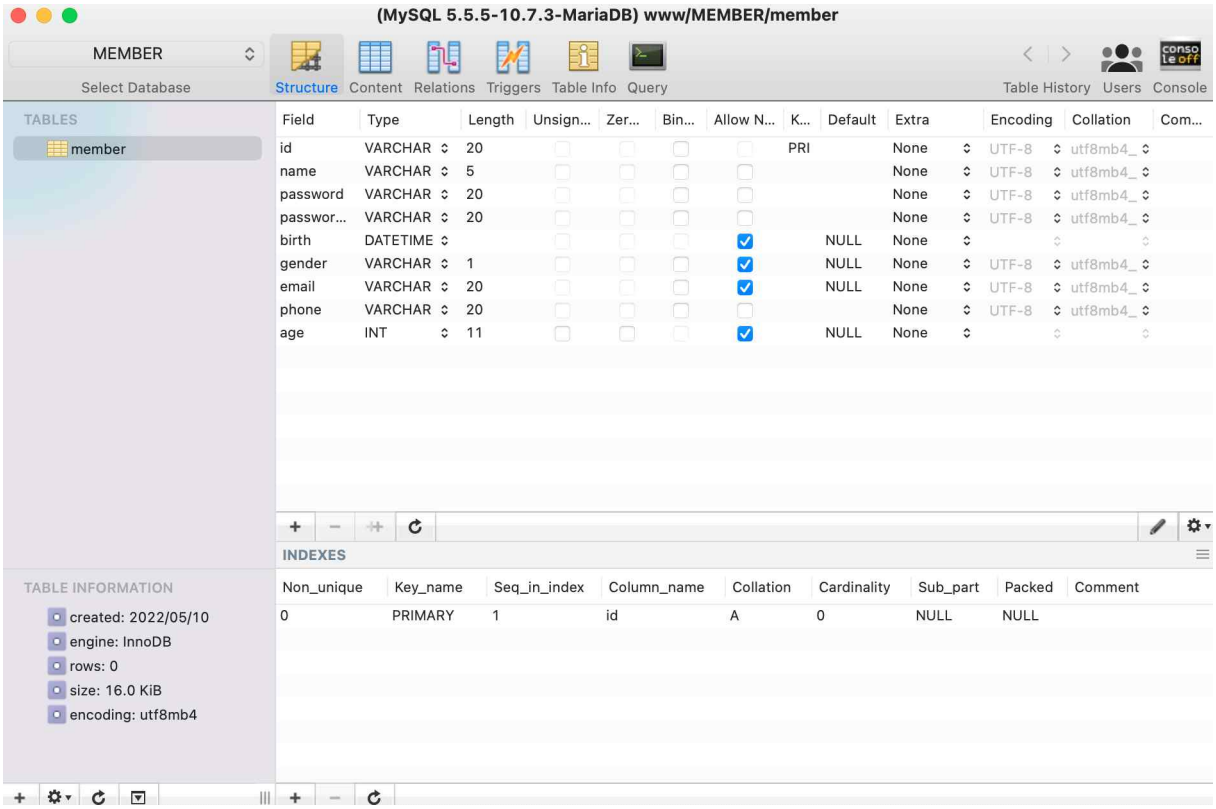

BACK
1. 회원정보 입력
2. 취향 선택
3. 가입완료
가입이 완료되었습니다.
다시 로그인 후
서비스를 즐겨주세요!
LOGIN


X
스물다섯
스물하나
2022 / 15+ / tvN
출연: 김태리, 남주혁, 김지연
LINK

10주차 : 회원 데이터베이스 생성

Mariadb를 이용하여 회원가입 정보를 받아 올 member table 생성

```
mysql> CREATE TABLE member (
    -> id varchar(20) primary key,
    -> name varchar(5) not null,
    -> password varchar(20) not null,
    -> passwordcheck varchar(20) not null,
    -> birth datetime ,
    -> gender varchar(1),
    -> email varchar(20),
    -> phone varchar(20) not null,
    -> age int
    -> );
Query OK, 0 rows affected (0.14 sec)
```

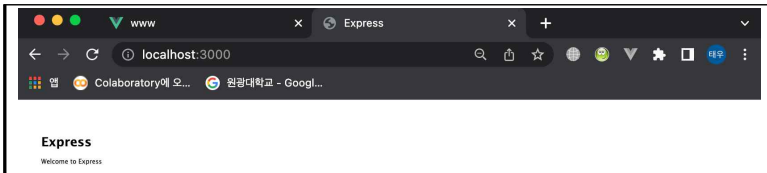


11주차 : 데이터베이스와 서버 연동 및 회원가입 페이지 개발

회원가입 페이지 개발

```
sudo npm install -g express-generator
```

```
express --view=pug backend
```



express 프레임워크 설치 서버환경 구축

12주차 알고리즘 선택

〈표 1〉 추천기법의 유형 구분

		사용자 성향	아이템 속성
사용기록 미활용		인구통계학적 규칙에 의한 추천	콘텐츠 기반 필터링
사용기록 활용	규칙 적용	-	연관규칙에 의한 추천
	협업 필터링	사용자 기반 협업 필터링	아이템 기반 협업 필터링

1) 인구통계학적 규칙에 의한 추천

전자상거래의 초기단계와 같이 사용기록(구매자와 구매리스트)이 없을 경우, 사용기록으로 사용자의 성향을 분석하여 추천하는 것은 불가능하다. 따라서 유사한 성향을 가진 사람들을 그룹화하여 개별 분석이 가능한 인구통계학적 필터링 추천을 할 수밖에 없다. 사용자가 입력한 개인정보(지역, 나이, 성별, 수입 등)에 따라 인구통계학적 특성(예를 들어, 서울, 부산, 울산 지역 등)으로 사용자들의 집단을 구분한다. 그리고 사용자 집단의 구성원이 특정한 아이템을 구매하면, 다른 구성원들에게도 해당 아이템을 추천하도록 추천대상으로 필터링한다. 예를 들어지역/나이/성별에 따라 구분된 사용자 집단 G의 한 사용자가 상품 A를 구매하면 집단 G에 속한 다른 사용자들에게 상품 A를 추천하는 방식이다.

2) 연관규칙에 의한 추천

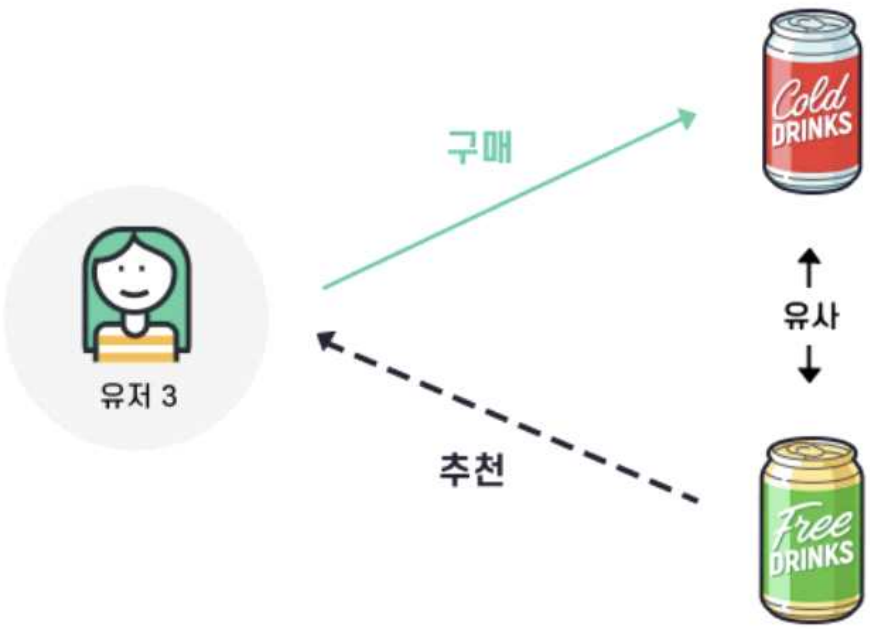
연관규칙(association rule)은 데이터들의 발생빈도를 사용하여 각 데이터 간의 연관관계를 밝히는

방법을 말한다. 예를 들면 슈퍼마켓에서 '시리얼'을 구매하는 고객들이 '우유'도 같이 구매하는 경우, 두 상품 간에는 연관관계가 있다고 분석하는 것과 같다. 이처럼 사용자의 구매빈도의 패턴에서 파악하는 연관관계의 가장 대표적인 기법이 'Apriori 알고리즘'이다.
Apriori 알고리즘을 이용한 추천의 과정을 세분하면 다음과 같다.

① 아이템들의 사용기록을 확보한다.
② 사용기록을 토대로 동시출현(동시구매) 빈도가 높은 빈발 아이템 집합(frequent item set)들을 판별하기 위해 지지도(support)를 계산한다.
③ 아이템 집합 간의 연관성의 강도를 측정하기 위해 신뢰도(confidence)를 계산한다.
④ 생성된 규칙이 실제 효용가치가 있는지를 판별하기 위해 향상도(lift)를 계산한다.
⑤ 동시구매의 패턴이 강한 연관규칙(지지도, 신뢰도, 향상도)을 가지는 아이템 집합을 찾아서 추천을 한다. 예를 들어, '시리얼'을 구매하는 고객은 '우유'를 동시에 구매하고 '라면'을 구매하는 고객은 '계란'과 '참치캔'을 동시에 구매하는 패턴이 나오면, 이들을 기반으로 '시리얼'을 구매하는 다른 고객들에게 '우유'를 추천하고, '라면'을 구매하는 고객들에게는 '계란'과 '참치캔'을 추천하는 방식이다.

Apriori 알고리즘은 구현하기 쉽고, 어느 정도 만족할 만한 결과를 보여준다. 그러나 아이템의 개수가 많아지면 계산의 복잡도가 엄청나게 증가하는 단점이 있다.

3) 콘텐츠 기반 필터링(Content-based Filtering, CBF)



콘텐츠(아이템)의 특성을 기술하는 메타정보를 기반으로 유사한 콘텐츠의 집단을 구분하여 추천하는 방식이다. 이러한 메타정보는 영화의 줄거리, 뉴스의 내용, 상품의 설명텍스트 등과 같이 대체로 텍스트 데이터에서 파악한다. 각 콘텐츠의 텍스트에서 속성정보를 가공해야 하는데, 속성정보는 주로 텍스트에서 추출한 단어리스트, 또는 단어가방(bag of words)으로 표현된다. 각 콘텐츠의 특성을 표

현하는 단어들(자질, features)을 식별하고, 단어들의 출현빈도를 확인하여, 이를 벡터(vector)로 표현한 것이다. 즉, 각 콘텐츠는 단어의 벡터로 표현하며, 그 벡터형태의 메타정보를 이용하여 사용자(수용자)는 원하는 아이템들을 선택하게 된다. 사용자가 원하는 바를 기술한 것을 사용자 프로파일이라 하며, 이 또한 단어들로 표현한다. 사용자 프로파일은 사용자가 사용(선호)했던 콘텐츠들에서 자주 나타나는 속성정보로 만들 수 있다. 아무튼 콘텐츠(내용) 기반 필터링은 사용자 프로파일의 단어들(사용자의 단어벡터)과 콘텐츠의 단어벡터들(콘텐츠 프로파일이라고도 함)과 유사성을 비교하여 적합한 콘텐츠들을 찾아내는 방식이다. 적합성의 정도에 따라 필터링된 결과의 순위화도 가능하다.

1) 개요
- 컨텐츠가 비슷한 아이템을 추천한다.
- 사용자가 과거에 경험했던 아이템 중 비슷한 아이템 추천
- 유저 A 가 높은 평점을 추거나 큰 관심을 갖은 아이템 X와 유사한 아이템 Y를 추천한다.

2) 중요점
- 컨텐츠의 특징들이 어떻게 이루어져 있는지 파악
- 특징들 간에 유사성을 파악
- 컨텐츠 분석하는 아이템 분석 알고리즘이 중요
- 선능을 높일 수 있는 적절한 컨텐츠를 사용

3) 장/단점
- 장점 :
  다른 유저의 데이터가 필요하지 않다.
  추천할 수 있는 아이템의 범위가 넓다. (다른 유저와 관계없이 추천 가능하다.)
  추천의 이유가 명확하다.
- 단점 :
  적절한 특징을 찾아야 성능이 올라간다.
  Cold Start, 새로운 유저를 위한 추천이 어렵다.
  유저 프로필 외에는 추천이 불가능해서, 선호하는 특성을 가진 항목을 반복 추천한다.

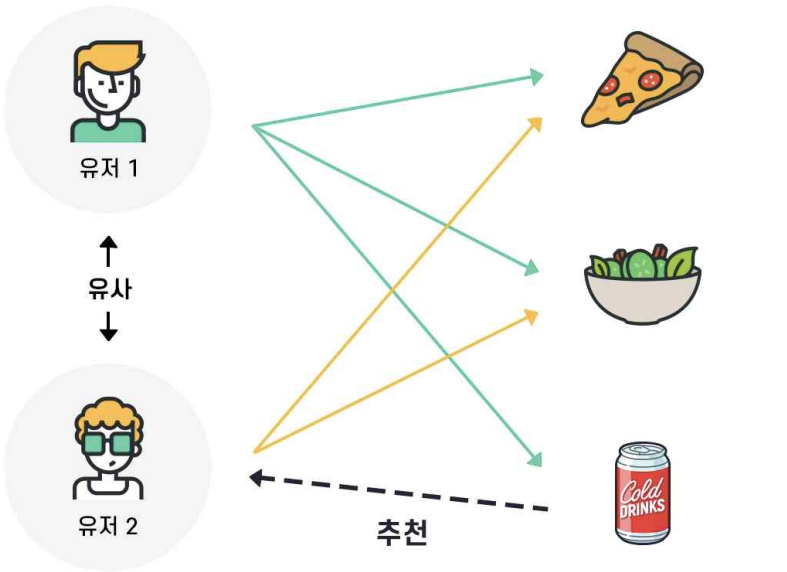
4) 알고리즘 과정
① 데이터 습득
② 컨텐츠 분석(Contents Analyer)
- 특징 추출
- Vector Representation : 벡터 집합으로 변경
③ 유저 프로필 파악 (Profile Learner)
④ 유사 아이템 선택 (Filtering Component)

콘텐츠 기반 필터링 추천의 장점은 사용기록이 없어도 정확도가 높은 추천이 가능하다는 점이며, 단점은 추천의 과정이 너무 복잡하다는 점이다.

4) 협업 필터링

협업 필터링(collaborative filtering; CF)에서 '협업'은 특정 사용자 집단의 유사한 사용행위를 의미

하며, '협업 필터링'은 협업에서 파악한 정보를 기반으로 추천대상을 추출하여 수요자에게 추천한다는 의미이다. 사용자 집단의 유사한 행위는 사용기록을 통해 파악한다. 협업 필터링은 **Amazon의 상품(도서) 추천, Netflix의 비디오 추천, GroupLens의 뉴스 추천, MovieFinder의 영화 추천 등에서 사용하는 알고리즘**이다. Amazon은 사용자의 상품평가(별점 또는 텍스트)데이터를 활용하여 연관성이 높은 상품을 추천한다. Netflix은 전 세계 수천만명의 고객이 비디오(영화) 감상 후 입력한 비디오 평가 데이터를 바탕으로 각 고객에 맞는 비디오를 추천한다. 협업 필터링은 사람들의 사용패턴을 사용하므로 사회 필터링(social filtering)이라고도 한다. 성향이 비슷한 사람들은 선호하는 것들도 비슷할 것이라는 가설을 전제로 한다.



1) 개요
- 많은 사람의 의견으로 더 나은 추천을 한다.
- 개인이 아닌 단체, 다수의 의견을 따른다.
- 나랑 같은 취향의 사람의 취향을 추천해 준다.

2) 과정
① 유저 A와 비슷한 평가를 한 유저 B 선정
② B가 호감을 나타낸 다른 컨텐츠 선정
③ 그 중 A와 성향이 비슷한 컨텐츠 추천

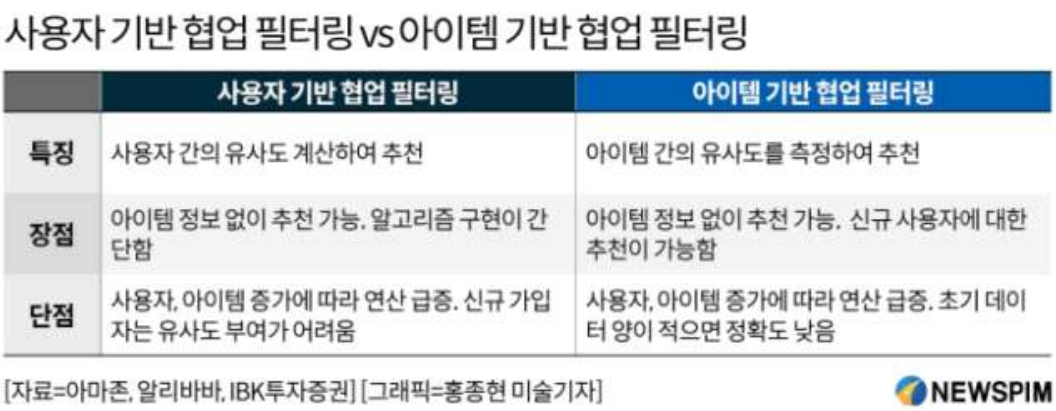
3) 종류
- 이웃기반 협업 필터링
: Item-based
: User-based
- 모델 기반 협업 필터링
: 딥러닝 협업 필터링

-하이브리드 협업 필터링

1) User-based
- 두 사용자가 얼마나 유사한 항목을 좋아했는지를 바탕으로 추천
- 플래폼에서 유저가 많은 부분을 차지할 때 사용
- SNS 등 사용자에 포커스 되어있는 서비스의 경우
- 아이템이 많을 시 = User based
- 특정 유저와 비슷한 유저로 분류된 실제 유저의 취향을 알 수 없다.
- 여러 유저의 데이터를 보았기 때문에 새로운 추천을 할 수 있다.

2) Item-based
- 아이템과 아이템 사이의 유사도 계산
- 과거 아이템 선호도를 기반으로 선호 연관성이 높은 다른 아이템 추천
- 컨텐츠 기반은 컨텐츠의 특성으로 추천, 협업 필터링은 과거의 선호도를 측정
- 아이템의 수가 크게 변하지 않는다면 Item-based
- 비슷한 아이템과 가중치로 함께 설명될 수 있다.
- 과거 아이템에 의존하기 때문에 새로운 아이템을 추천하기 어렵다.

3) 문제점
- Cold Start 문제 : 충분한 데이터가 필요하다.
- 계산 량
: 데이터가 많아질수록 유사도 계산이 많아진다.
: 그러나 데이터가 많아야 추천 품질이 높아진다.
- Long Tail Economy
: 대부분의 사용자가 관심가지는 소수의 아이템으로 추천이 쏠릴 수 있다.
: 관심이 부족한 아이템은 추천되지 못한다.



13주차 알고리즘 연구

컨텐츠 기반 필터링 구현 프로세스

1. 컨텐츠에 대한 여러 텍스트 정보들을 피처 벡터화

2. 코사인 유사도로 컨텐츠별 유사도 계산
3. 컨텐츠 별로 가중 평점을 계산
4. 유사도가 높은 컨텐츠 중에 평점이 좋은 컨텐츠 순으로 추천

데이터셋 : Kaggle TMDB 5000 Movie Dataset - movie.csv

```
movies = pd.read_csv("/content/drive/MyDrive/datasets/tmdb_5000_movies.csv")
movies.head(1)
```

데이터 셋을 가져와 movies 변수에 저장

위 그림을 보아 22개의 속성값을 가진 데이터를 볼 수 있음

```
movies.shape
```

```
(4803, 20)
```

movies.shape을 통해 4803개의 데이터를 가지고 있는 것을 알 수 있음.

```
# 주요 컬럼으로 데이터 프레임 생성
col_lst = ['id', 'title', 'genres', 'vote_average', 'vote_count', 'popularity', 'keywords', 'overview']
movies_df = movies[col_lst]
```

필요한 속성(collum)값만 추출하여 새로운 데이터 테이블을 만듬.

- id: 아이디
- title: 영화 제목
- genres: 영화가 속한 여러가지 장르
- vote_average: 평균 평점
- vote_count: 평점 투표 수
- popularity: 영화의 인기 정도
- keywords: 영화를 설명하는 주요 키워드 문구

- overview: 영화에 대한 개요 설명

```
from ast import literal_eval

movies_df['genres'].apply(literal_eval)[0]
```

```
[{'id': 28, 'name': 'Action'},
 {'id': 12, 'name': 'Adventure'},
 {'id': 14, 'name': 'Fantasy'},
 {'id': 878, 'name': 'Science Fiction'}]
```

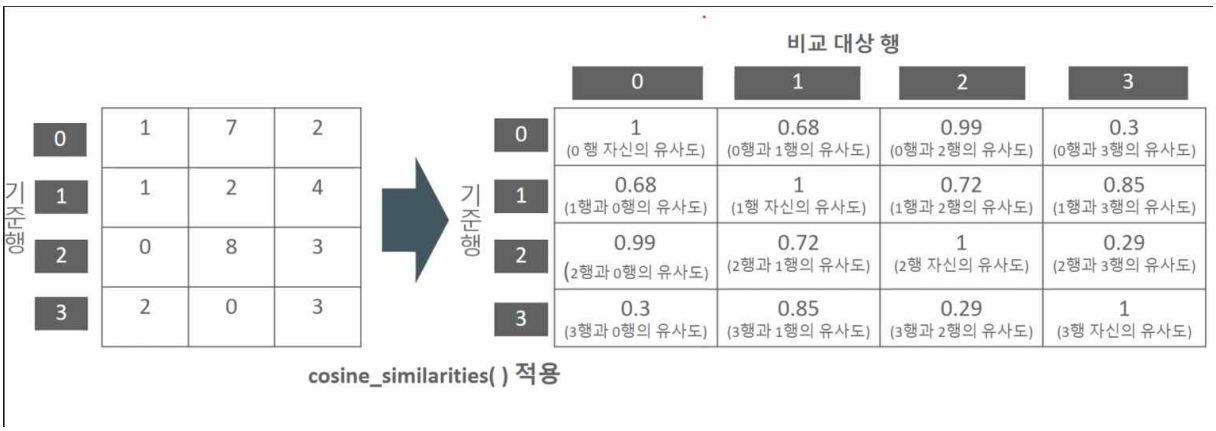
movies_df에 저장된 데이터들의 name만 추출.

```
from ast import literal_eval

# 문자열을 객체로 변경: 리스트 내의 사전
movies_df['genres'] = movies_df['genres'].apply(literal_eval)
movies_df['keywords'] = movies_df['keywords'].apply(literal_eval)

# 객체에서 name만 추출: 사전 마다 name을 추출
movies_df['genres'] = movies_df['genres'].apply(lambda x : [ dic['name'] for dic in x] )
movies_df['keywords'] = movies_df['keywords'].apply(lambda x : [ dic['name'] for dic in x] )

movies_df[['genres', 'keywords']][:1]
```




```
from sklearn.feature_extraction.text import CountVectorizer

# 리스트 객체를 문자열로 변환: 공백으로 구분
movies_df['genres_literal'] = movies_df['genres'].apply(lambda x : (' ').join(x))

# CountVectorizer
count_vect = CountVectorizer(min_df=0, ngram_range=(1,2))
genre_mat = count_vect.fit_transform(movies_df['genres_literal'])

print(genre_mat.shape)
```

```
(4803, 276)
```

cosine_similarity() 함수를 사용하여 유사도 측정

```
from sklearn.metrics.pairwise import cosine_similarity

genre_sim = cosine_similarity(genre_mat, genre_mat)
genre_sim[0]
```

```
array([1.        , 0.59628479, 0.4472136 , ..., 0.        , 0.        ,
       0.        ])
```

입력한 영화의 유사도를 측정하여 내림차수로 표현해주는 함수를 생성

```
def find_sim_movie(df, sim_matrix, title_name, top_n=10):

    # 입력한 영화의 index
    title_movie = df[df['title'] == title_name]
    title_index = title_movie.index.values

    # 입력한 영화의 유사도 데이터 프레임 추가
    df["similarity"] = sim_matrix[title_index, :].reshape(-1,1)

    # 유사도 내림차순 정렬 후 상위 index 추출
    temp = df.sort_values(by="similarity", ascending=False)
    final_index = temp.index.values[ : top_n]

    return df.iloc[final_index]
```

'대부'라는 영화와 장르별 유사도가 높은 영화 10개를 추출

```
# The Godfather(대부)와 장르별 유사도가 높은 영화 10개
similar_movies = find_sim_movie(movies_df, genre_sim, 'The Godfather', 10)
similar_movies[['title', 'vote_average', "similarity"]]
```

	title	vote_average	similarity
3636	Light Sleeper	5.7	1.0
892	Casino	7.8	1.0
3866	City of God	8.1	1.0
1243	Mean Streets	7.2	1.0
1370	21	6.5	1.0
4041	This Is England	7.4	1.0
1847	GoodFellas	8.2	1.0
2582	The Place Beyond the Pines	6.8	1.0
1946	The Bad Lieutenant: Port of Call - New Orleans	6.0	1.0
4217	Kids	6.8	1.0

가중 평점 함수를 생성

```
# 평가 횟수와 평점을 모두 고려한 가중 평점 함수를 생성

percentile = 0.6
m = movies_df['vote_count'].quantile(percentile)
C = movies_df['vote_average'].mean()

def weighted_vote_average(record):
    v = record['vote_count']
    R = record['vote_average']

    return ( (v/(v+m)) * R ) + ( (m/(m+v)) * C )

movies_df['weighted_vote'] = movies_df.apply(weighted_vote_average, axis=1)
```

장르 유사도와 가중 평점을 모두 고려한 영화 추천 함수

```python
# 장르 유사도와 가중 평점을 모두 고려한 영화 추천 함수
def find_sim_movie(df, sim_matrix, title_name, top_n=10):

    # 입력한 영화의 index
    title_movie = df[df['title'] == title_name]
    title_index = title_movie.index.values

    # 입력한 영화의 유사도 데이터 프레임 추가
    df["similarity"] = sim_matrix[title_index, :].reshape(-1,1)

    # 유사도와 가중 평점순으로 높은 상위 index 추출 (자기 자신 제거)
    temp = df.sort_values(by=["similarity", "weighted_vote"], ascending=False)
    temp = temp[temp.index.values != title_index]

    final_index = temp.index.values[:top_n]

    return df.iloc[final_index]
```

함수를 적용하여 'The Godfather'라는 영화와 유사한 장르 중 평점이 높은 10개의 영화를 추출

```
similar_movies = find_sim_movie(movies_df, genre_sim, 'The Godfather', 10)
similar_movies[['title', 'vote_average', "weighted_vote", "similarity"]]
```

	title	vote_average	weighted_vote	similarity
1881	The Shawshank Redemption	8.5	8.396052	1.0
2731	The Godfather: Part II	8.3	8.079586	1.0
1847	GoodFellas	8.2	7.976937	1.0
3866	City of God	8.1	7.759693	1.0
1663	Once Upon a Time in America	8.2	7.657811	1.0
3887	Trainspotting	7.8	7.591009	1.0
883	Catch Me If You Can	7.7	7.557097	1.0
892	Casino	7.8	7.423040	1.0
281	American Gangster	7.4	7.141396	1.0
4041	This Is England	7.4	6.739664	1.0

구분	일자	사용 내역	금액
합계			

※ 최종 결과보고서에는 반드시 개발 작품의 사진이 포함되어야 함