

Denim.

8조

15113326 신호준
16102186 명인지
17101934 고세윤
17101986 조기혁
20102040 이은재

목차

001 프로젝트 목적

002 크롤링

003 DB

004 웹서버 구축

005 알고리즘



Part 1

프로젝트 목적

프로젝트 목적

- 프로젝트 주제선정 동기
 - 쇼핑몰 상품 사진과 실착의 차이를 줄일 수 있는가?
 - 2021 fashion hot item은 청바지
- 프로젝트 목적
 - 고객의 체형에 맞는 바지 추천 시스템
 - 고객의 니즈에 맞는 바지 추천 시스템
 - 쇼핑몰 상품 사진과 실착의 차이 최소화
- 크롤링 사이트
 - 무신사

The image shows the MUSINSA logo, which consists of the word "MUSINSA" in a white, bold, sans-serif font centered on a solid black rectangular background.

Part 2

크롤링

크롤링

크롤링 할 Product Data

Product

한정 판매 1967 JET BLACK JEANS [WIDE STRAIGHT]



한정 판매

무신사 스토어를 포함한 제한된 판매처에서만 구매 가능한 상품입니다.

Product Info 제품정보

브랜드 / 품번 **BRANDED / LOT1967**
 시즌 ⓘ / 성별 **2020 F/W / 남**
 조회수(1개월) **31.6만 회 이상**
 누적판매(1년) **2.6만 개 이상 (결제완료-반품)**
 좋아요 **42,424**
 구매 후기 ⓘ **★★★★★ 4.8 / 후기 29,731개 보기**

#청바지 #블랙진 #워싱턴 #와이드 #데님팬츠 #와이드데님
 #흑청

19~28세, 남성에게 인기 많은 상품

Delivery Info 배송정보

배송 방법 **국내 배송 / 임점사 배송 / CJ대한통운**
 출고 기간 ⓘ **평균 0.4일 / 주말, 공휴일 제외**

Price Info 가격정보

무신사 판매가 **78,000원**
 무신사 회원가 **64,940 ~ 68,000원** ∨ **↓ 13% 이상 할인**
 무이자 할부가 **8,118 ~ 8,500원 (8개월)**
 무신사 적립금 **최대 2,720원**

한정 판매

무신사 스토어를 포함한 제한된 판매처에서만 구매 가능한 상품입니다.

Product Info 제품정보

브랜드 / 품번 **BRANDED / LOT1967**
 시즌 ⓘ / 성별 **2020 F/W / 남**
 조회수(1개월) **31.7만 회 이상**
 누적판매(1년) **2.6만 개 이상 (결제완료-반품)**
 좋아요 **42,658**
 구매 후기 ⓘ **★★★★★ 4.8 / 후기 29,898개 보기**

#청바지 #블랙진 #워싱턴 #와이드 #데님팬츠 #와이드데님
 #흑청

Price Info 가격정보

무신사 판매가 **78,000원**
 무신사 회원가 **64,940 ~ 68,000원** ∨ **↓ 13% 이상 할인**
 무이자 할부가 **8,118 ~ 8,500원 (8개월)**
 무신사 적립금 **최대 2,720원**

크롤링

크롤링 할 Review Data

Review

LV 4 Taeng 1

남성, 168cm, 60kg · 신고



1880 PURE CREAM JEANS [WIDE STRAIGHT]
28 구매



튀지 않으면서 무난하게 이쁜 크림진 찾고 있었는데 너무 최고네요!
무엇보다 코디의 범용성이 정말 넓어 꼭 하나 있으면 좋은 바지라 생각이 되네요!

사이즈 보통이에요

밝기 보통이에요

색감 선명해요

두께감 보통이에요



👍 도움돼요 1

😊 스타일 좋아요 1

남성, 168cm, 60kg · 신고



1880 PURE CREAM JEANS [WIDE STRAIGHT]
28 구매



튀지 않으면서 무난하게 이쁜 크림진 찾고 있었는데 너무 최고네요!
무엇보다 코디의 범용성이 정말 넓어 꼭 하나 있으면 좋은 바지라 생각이 되네요!

사이즈 보통이에요

밝기 보통이에요

색감 선명해요

두께감 보통이에요



👍 도움돼요 1

😊 스타일 좋아요 1

크롤링

▪ Product, Review, Product_age, Age Table

• Product

Product_id	name	Hashtag	Rating	Image	Product_total	View_total	Price	Heart	Ratew	Tpw
상품번호	상품명	해쉬태그	별점	상품 이미지	총 구매수	총 조회수	가격	좋아요	별점 가중치	총 구매수 가중치

• Review

Review_id	Product_id	Name	Gender	Height	Weight	Rating	Fit_size	evls	evlb	evlc	evlt	link
리뷰번호	상품번호	Reviewer의 키	Reviewer의 몸무게	상품명	Reviewer가 내린 별점	구매한 바지 사이즈	사이즈	사이즈에 대한 평가척도	밝기에 대한 평가 척도	컬러에대한 평가척도	두께감에 대 한 평가척도	링크

• Product_age

Product_id	Agegroup	Apw
상품번호	연령대	연령별 구매 가중치

• Age

Agegroup
연령대

크롤링

▪ Python Code - product

```
i = 0
for jean in jean_list:
    try:
        jean.update(heart=heart[i])
        i += 1
    except:
        jean.update(heart=0)
        print("오류로 인해 좋아요에 0을 입력하였습니다, 계속 진행 중 취소하지 마세요.")

# json 데이터 출력
with open('musinsa_jeans.json', 'w', encoding="utf-8") as f:
    f.write(json.dumps(jean_list,ensure_ascii=False))

print(time.time() - start)
print("완료")

except:
    print("에러발생 - 작업 저장 중")
    with open('musinsa_jeans.json', 'w', encoding="utf-8") as f:
        f.write(json.dumps(jean_list,ensure_ascii=False))

    err = traceback.format_exc()
    ErrorLog(str(err))
    print(time.time() - start)
```

크롤링

▪ Python Code - product

```
# 상품번호로 상품 정보 가져오기 시작
try:
    for jean in jean_list:
        breaknum += 1
        if(breaknum % 10 == 0):
            sleep(random.uniform(8, 9))
            print(str(jean)+"breaking")

    url = "https://store.musinsa.com/app/goods/"+str(jean['num'])
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36'+
        ' (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36'}
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, "lxml")

    # 상품 이름
    name = soup.select('span.product_title em')
    a = re.sub('<.+?>', '', str(name), 0, 0).strip()
    name = a[1:-1]
    jean.update(name=name)

    # 상품 브랜드
    brand = soup.select('p.product_article_contents strong a')
    a = re.sub('<.+?>', '', str(brand), 0, 0).strip()
    brand = a[1:-1]
    jean.update(brand=brand)

    # 상품 해시태그
    tag = soup.select('p.product_article_contents a.listItem')
    a = re.sub('<.+?>', '', str(tag), 0, 0).strip()
    tag = re.sub(',', ' ', a, 0, 0).strip().replace(" ", '')
    tag = tag[1:-1]
    jean.update(tag=tag)
```

크롤링

▪ Python Code - product

```
i = 0
for jean in jean_list:
    try:
        jean.update(heart=heart[i])
        i += 1
    except:
        jean.update(heart=0)
        print("오류로 인해 좋아요에 0을 입력하였습니다, 계속 진행 중 취소하지 마세요.")

# json 데이터 출력
with open('musinsa_jeans.json', 'w', encoding="utf-8") as f:
    f.write(json.dumps(jean_list,ensure_ascii=False))

print(time.time() - start)
print("완료")

except:
    print("에러발생 - 작업 저장 중")
    with open('musinsa_jeans.json', 'w', encoding="utf-8") as f:
        f.write(json.dumps(jean_list,ensure_ascii=False))

err = traceback.format_exc()
ErrorLog(str(err))
print(time.time() - start)
```

크롤링

Python Code - review

```
# print(body)
name = soup.select('div.review-list a.review-goods-information_name')[i]
name = re.sub('<.+?>', '', str(name), 0, 0).strip().replace(" ", "")
# print(name)
size = soup.select('div.review-list span.review-goods-information_option')[i]
size = re.sub('<.+?>', '', str(size), 0, 0).strip().replace(" ", "")
a = size.find(',')
if(a != -1):
    size = size[:a-1]
else:
    size

a = str(soup.select('div.review-list span.review-list__rating_unit')[i])
#print(a)
b= a.find('width: ')
c= a.find("></span>")
rating = float("{:1f}".format(float(str(a[b+7:c]))[:1])/20))

evl = soup.select('div.review-list ul.review-evaluation_list')[i]
evl = re.sub('<.+?>', '', str(evl), 0, 0).strip().replace(" ", "").split('\n')
evls=int({len(evl[0])==5 : 3, len(evl[0])==8: 2 }.get(True,1))
evlb=int({len(evl[1])==5 : 3, len(evl[1])==7: 2 }.get(True,1))
evlc=int({len(evl[2])==6 : 3, len(evl[2])==7: 2 }.get(True,1))
evlt=int({len(evl[3])==7 : 3, len(evl[3])==8: 2 }.get(True,1))

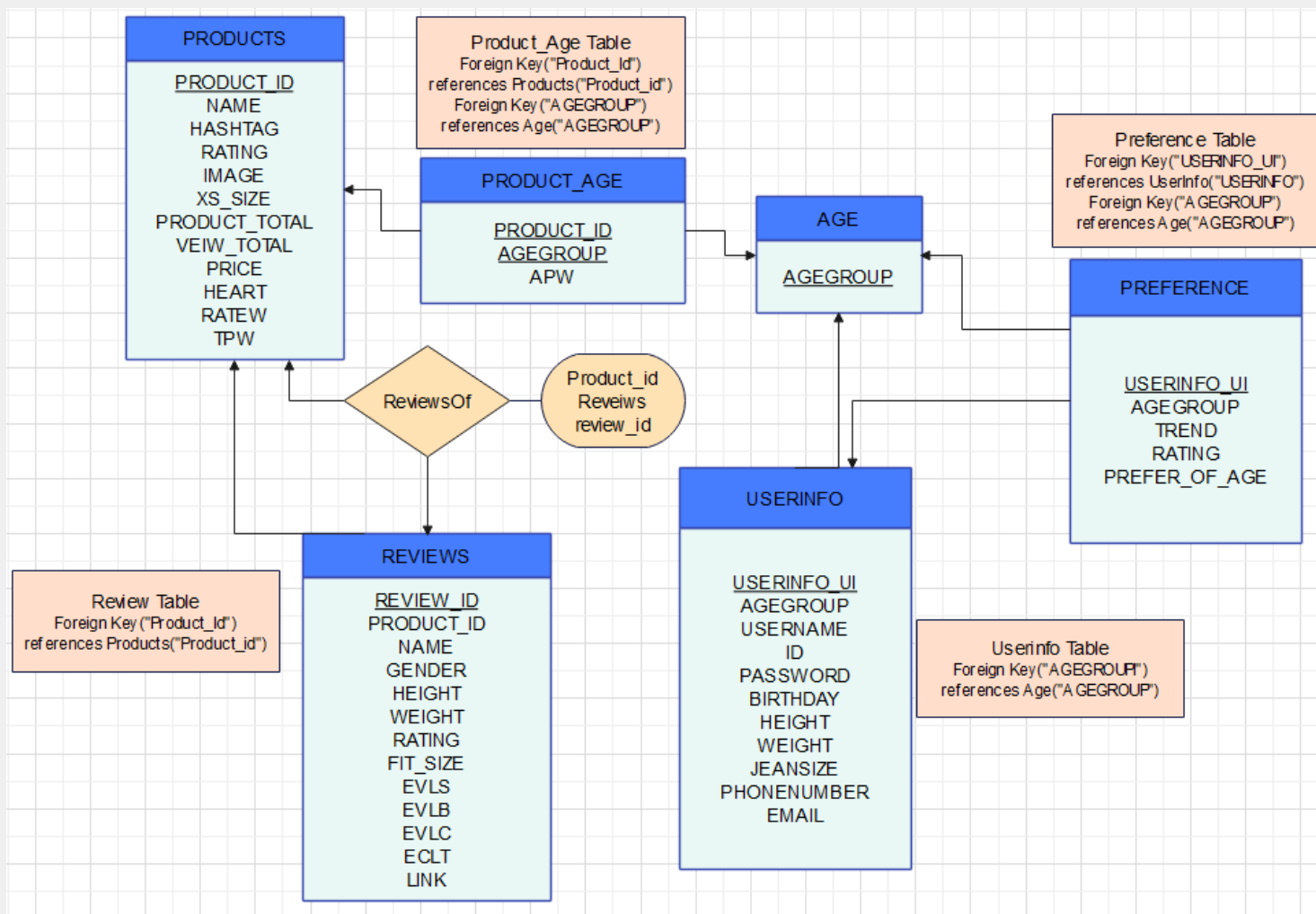
# print(evl)
link = str(soup.select('div.review-list li.review-content-photo__item img')[i])
a = int(link.find("src"))+5
b = -3
link = link[a:b]
review.append({"num": jean_num, "gender":gender, "height":height,"weight":weight, "name": name, "rating": rating, "size": size,
| "evls": evls,"evlb": evlb, "evlc": evlc, "evlt": evlt, "link": link})
```

Part 3

DB

DB

▪ ER diagram



DB

- SQL 코드
 - Table 생성 (Products, Products_age)

```
CREATE TABLE "PRODUCTS"(  
  "PRODUCT_ID" NUMBER(10,0) PRIMARY KEY,  
  "NAME" VARCHAR2(100),  
  "HASHTAG" VARCHAR2(200),  
  "RATING" NUMBER(2,1),  
  "IMAGE" VARCHAR2(200),  
  "XS_SIZE" VARCHAR2(10),  
  "PURCHASE_TOTAL" VARCHAR2(100),  
  "VEIW_TOTAL" VARCHAR2(100),  
  "PURCHASE_AGE" VARCHAR2(100),  
  "VEIW_AGE" VARCHAR2(100),  
  "PRICE" NUMBER(10,0),  
  "HEART" NUMBER(7,0),  
  "RATEW" NUMBER(2,1),  
  "TPW" NUMBER(2,1)  
);
```

```
create table "PRODUCTS_AGE"(  
  "PRODUCT_ID" NUMBER(10,0),  
  "AGEGROUP" NUMBER(1),  
  "APW" NUMBER(2,1),  
  CONSTRAINT PID PRIMARY KEY("PRODUCT_ID","AGEGROUP"),  
  FOREIGN KEY("PRODUCT_ID") REFERENCES PRODUCTS("PRODUCT_ID"),  
  FOREIGN KEY("AGEGROUP") REFERENCES AGE("AGEGROUP")  
);
```

Product_id	Name	HashTag	Rating	Image	XS_size	Purchase_total
number	var char2	varchar2	number	varchar 2	varchar2	varchar2
View_total	Purchase_age	View_age	Price	Heart	Ratew	Tpw
varchar2	varchar2	varchar2	number	number	number	number

Product_id	Agegroup	Apw
number	number	number

DB

- SQL 코드
 - Table 생성 (Reviews, Reviewed_of, Age)

```
CREATE TABLE "REVIEWS"(  
  "REVIEW_ID" NUMBER(7,0) PRIMARY KEY,  
  "PRODUCT_ID" NUMBER(10,0),  
  "NAME" VARCHAR2(100),  
  "GENDER" VARCHAR2(10),  
  "HEIGHT" NUMBER(4,1),  
  "WEIGHT" NUMBER(4,1),  
  "RATING" NUMBER(2,1),  
  "FIT_SIZE" VARCHAR2(4),  
  "EVLS" NUMBER(3,0),  
  "EVLB" NUMBER(3,0),  
  "EVLG" NUMBER(3,0),  
  "EVLH" NUMBER(3,0),  
  "LINK" VARCHAR2(1000),  
  FOREIGN KEY("PRODUCT_ID") REFERENCES PRODUCTS("PRODUCT_ID")  
);
```

```
CREATE TABLE "REVIEWED_OF"(  
  "PRODUCT_ID" NUMBER(10,0),  
  "REVIEW_ID" NUMBER(7,0),  
  FOREIGN KEY("PRODUCT_ID") REFERENCES "PRODUCTS",  
  FOREIGN KEY("REVIEW_ID") REFERENCES "REVIEWS"  
);
```

```
CREATE TABLE "AGE"(  
  "AGEGROUP" NUMBER(1) PRIMARY KEY  
);
```

Review_id	Product_id	Name	Gender	Height	Weight	
number	number	varchar2	varchar2	number	number	
Rating	Fit_size	Evls	Evlb	Evlc	Evlh	Link
number	varchar2	number	number	number	number	varchar2

Product_id	Review_id
number	number

Agegroup
number

Part 4

웹서버 구축

웹서버 구축

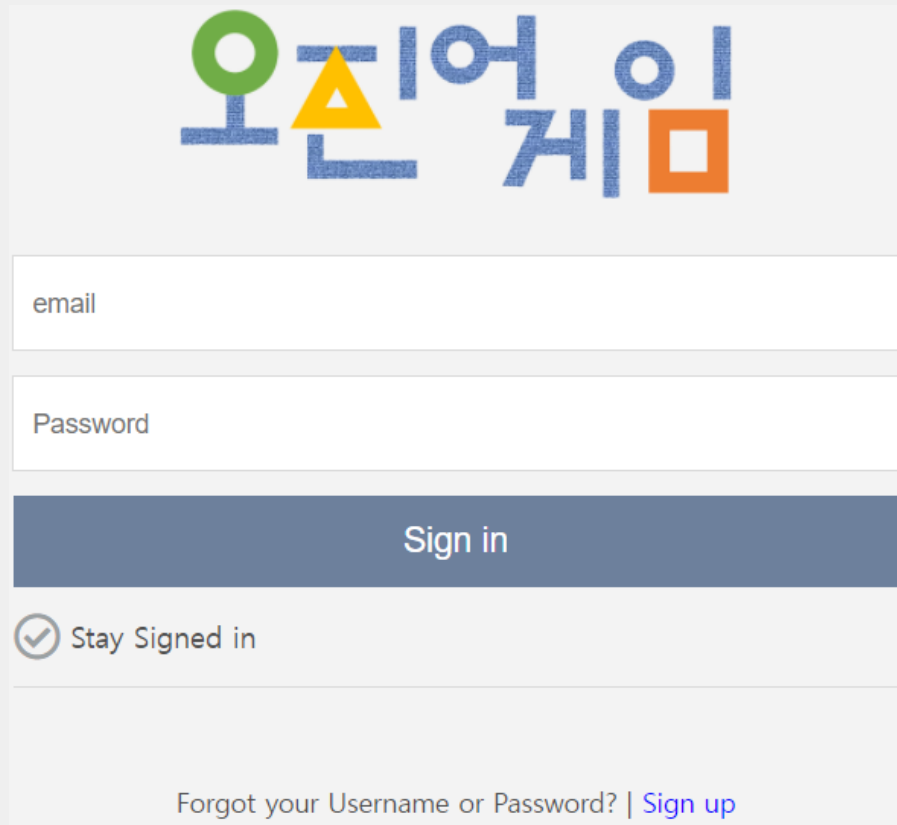
- 웹페이지 시각화
- 첫화면



웹서버 구축

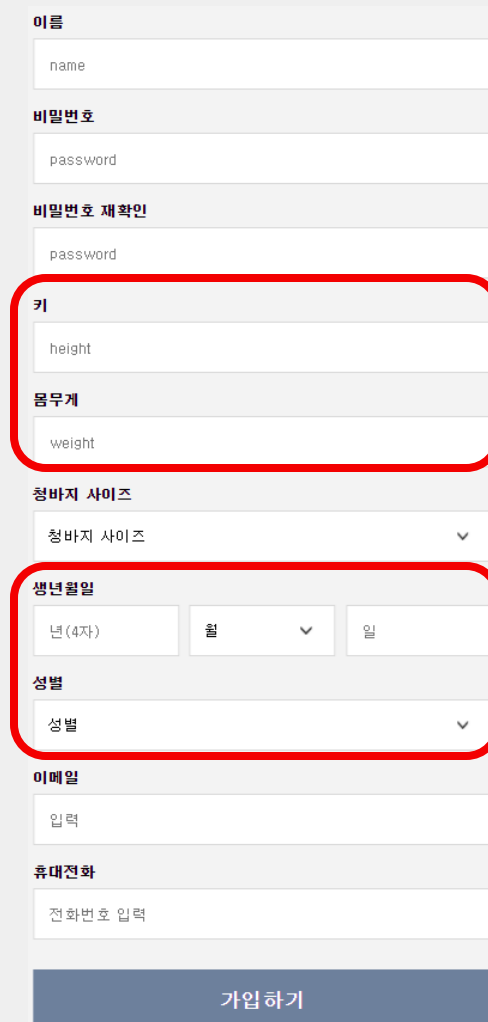
- 웹페이지 시각화
 - 로그인페이지, 회원가입페이지

- 로그인 페이지



The login page features a large, stylized logo at the top that reads '오인어임' (O-in-eo-im) in green, blue, and orange. Below the logo is a form with two input fields: 'email' and 'Password'. A dark blue 'Sign in' button is positioned below the password field. Underneath the button is a checkbox labeled 'Stay Signed in' with a checkmark icon. At the bottom, there is a link that says 'Forgot your Username or Password? | Sign up'.

- 회원가입 페이지



The sign-up page contains a series of form fields for user registration. The fields are: '이름' (name), '비밀번호' (password), '비밀번호 재확인' (password confirmation), '키' (height), '몸무게' (weight), '청바지 사이즈' (jeans size), '생년월일' (date of birth), '성별' (gender), '이메일' (email), and '휴대전화' (mobile phone). The '키' and '몸무게' fields are grouped together and highlighted with a red border. The '생년월일' field is also highlighted with a red border and includes sub-fields for year (year (4 digits)), month, and day. The '성별' field is a dropdown menu. At the bottom, there is a dark blue button labeled '가입하기' (sign up).

웹서버 구축

- 웹페이지 시각화
- 고객 니즈 파악 페이지



Q1. ABOUT MY FAVORITE JEAN

주어진 4가지 요소 (트렌트, 연령, 별점)를 자신이 좋아하는 jeans를 고를 때 중요하다고 생각하는 순서대로 체크해주세요!

trend

agegroup

rating

초기화



Q1. ABOUT MY FAVORITE JEAN

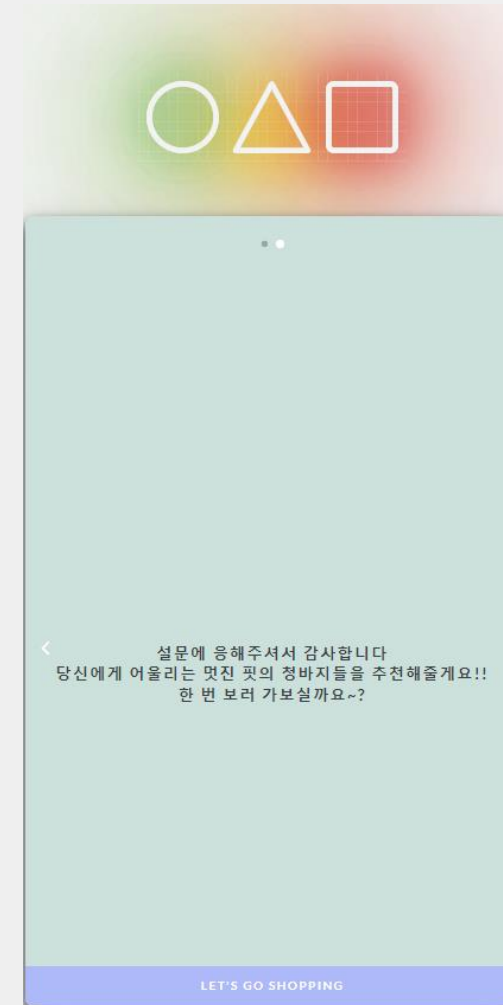
주어진 4가지 요소 (트렌트, 연령, 별점)를 자신이 좋아하는 jeans를 고를 때 중요하다고 생각하는 순서대로 체크해주세요!

trend 1

agegroup 3

rating 2

초기화



< 설문에 응해주셔서 감사합니다
당신에게 어울리는 멋진 핏의 청바지들을 추천해줄게요!!
한 번 보러 가보실까요~?

LET'S GO SHOPPING

웹서버 구축

- 웹페이지 시각화
- 고객 니즈 파악 정보 Table

- Userinfo

Userinfo_ui	Agegroup	Username	Id	Password	Birthday	Height	Weight	Jeansize	Phonenumber	Email
유저정보 번호	연령대	유저이름	아이디	비밀번호	생년월일	키	몸무게	바지 사이즈	전화번호	이메일

- Preference

Userinfo_ui	Agegroup	Trend	Rating	Prefer_of_age
유저정보 번호	연령대	트렌드	별점	연령에 따른 선호 바지

웹서버 구축

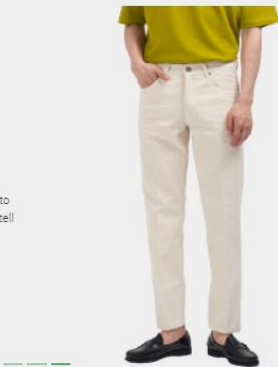
- 웹페이지 시각화
- 메인페이지



Repr in voluptate

Ullamco laboris nisi ut

We bring you 100% free CSS templates for your websites. If you wish to support TemplateMo, please make a small contribution via PayPal or tell your friends about our website. Thank you.



Categories of The Month

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Slim

Go Shop



Wide

Go Shop



Tapered

Go Shop

Featured Product

Reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident.



★★★★★

\$240.00

Gym Weight

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sunt in culpa qui officia deserunt.

Reviews (24)



★★★★★

\$480.00

Cloud Nike Shoes

Aenean gravida dignissim finibus. Nullam ipsum diam, posuere vitae pharetra sed, commodo ullamcorper.

Reviews (48)



★★★★★

\$360.00

Summer Addides Shoes

Curabitur ac mi sit amet diam luctus porta. Phasellus pulvinar sagittis diam, et scelerisque ipsum lobortis nec.

Reviews (74)

Zay Shop

123 Consectetur at ligula 10660
010-020-0340
info@company.com

Products

Luxury
Sport Wear
Men's Shoes
Women's Shoes
Popular Dress
Gym Accessories
Sport Shoes

Further Info

Home
About Us
Shop Locations
FAQs
Contact

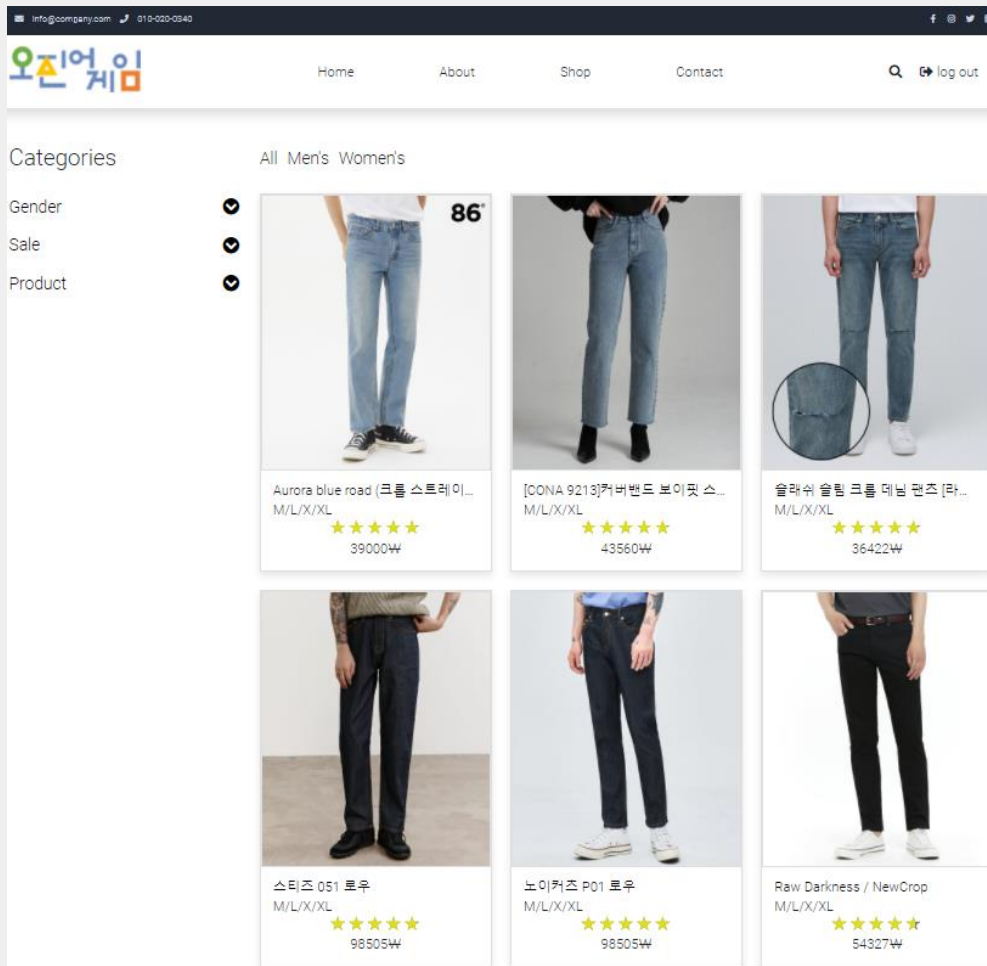


Email address

Subscribe

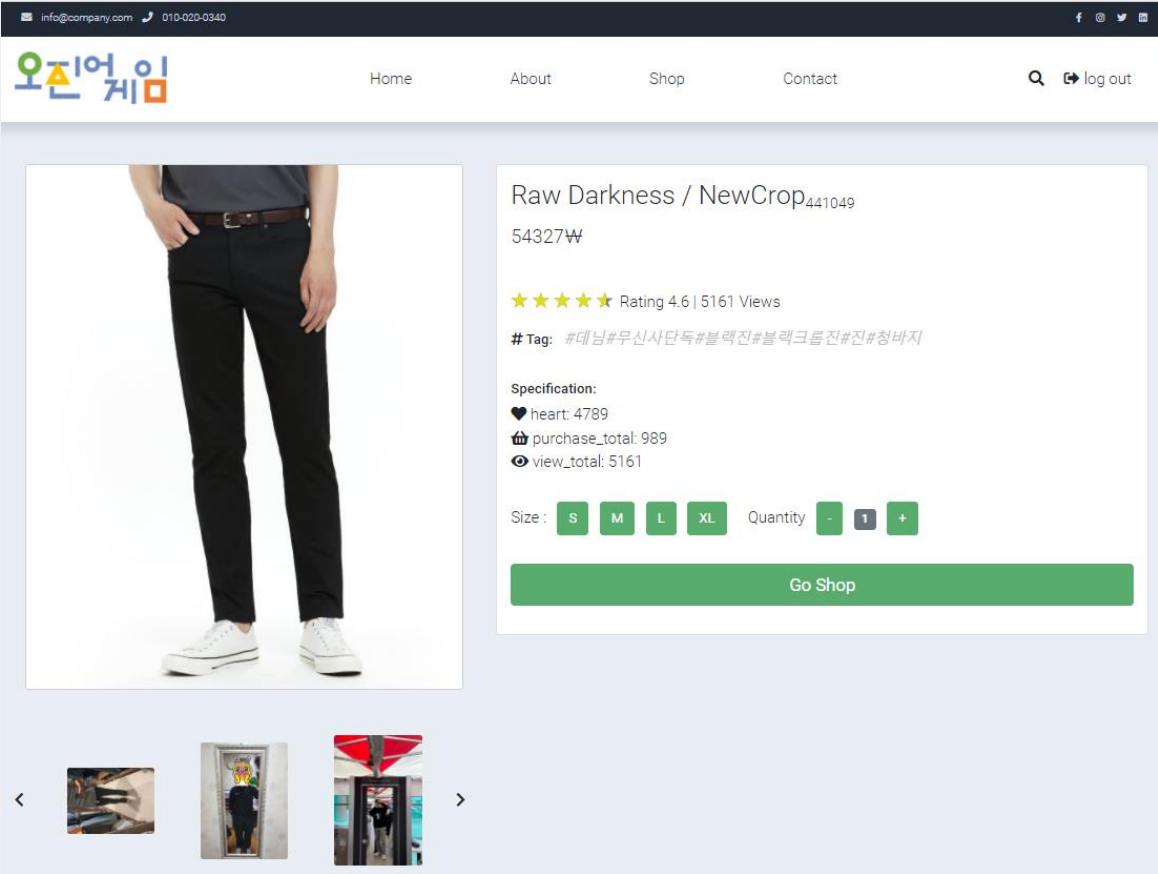
웹서버 구축

- 웹페이지 시각화
- 상품페이지
- 상품 리스트 페이지(3페이지)

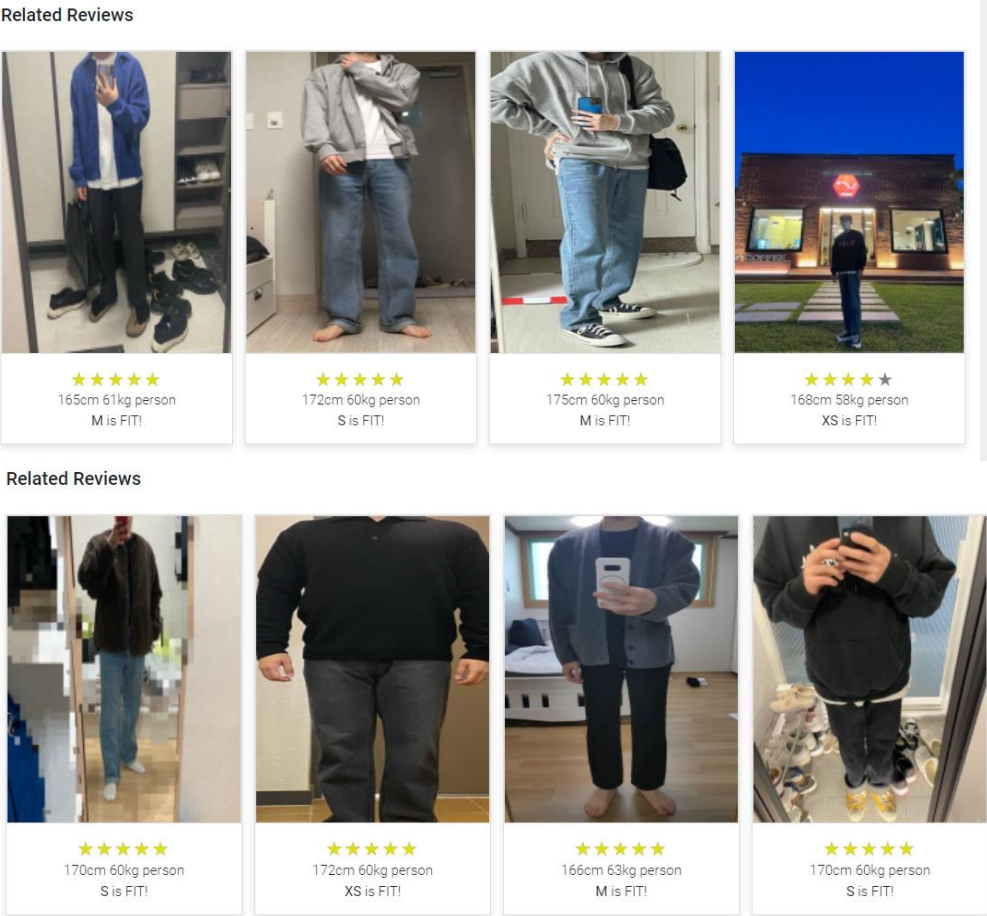


웹서버 구축

- 웹페이지 시각화
- 상품페이지
- 상품 상세 페이지



- 상품 상세 페이지- 관련 리뷰

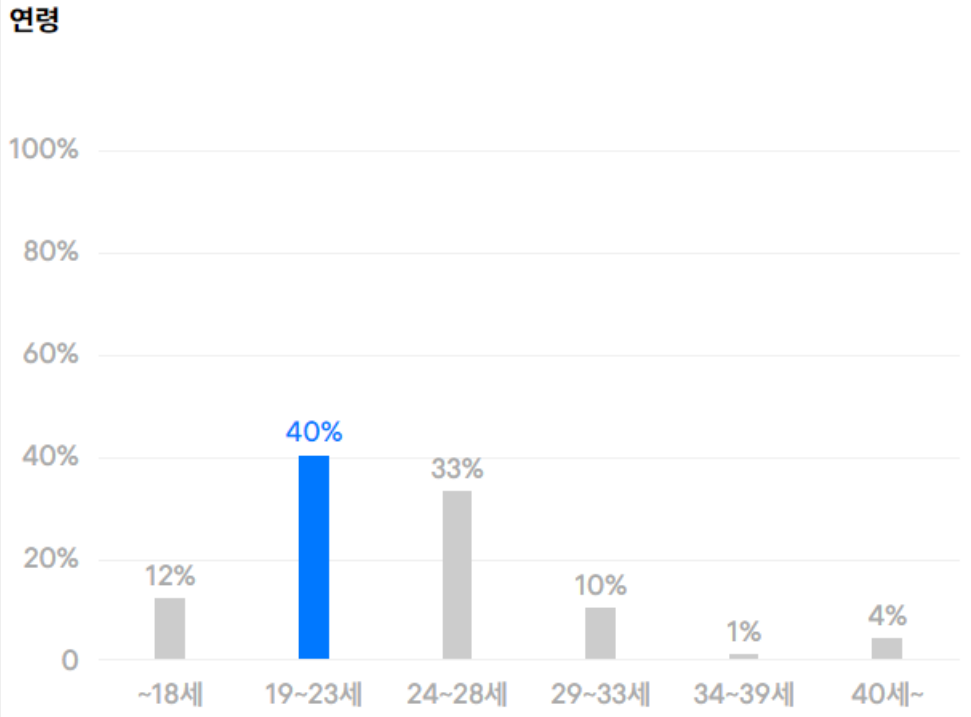
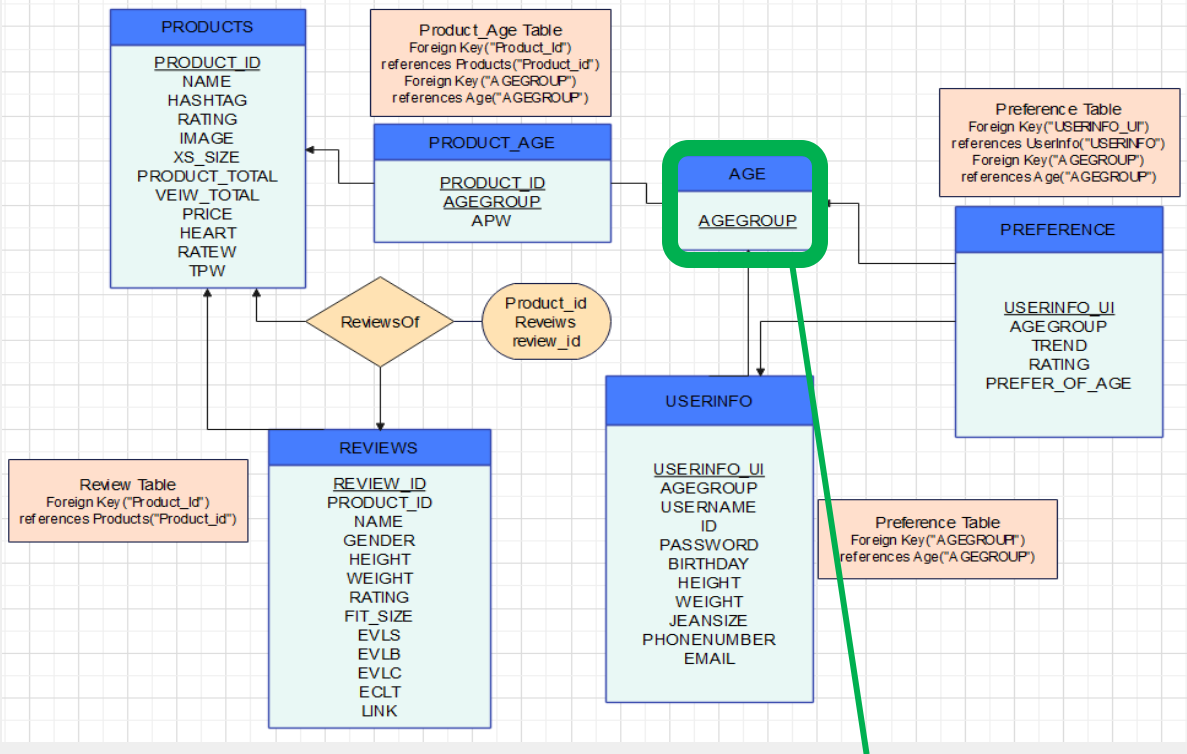


Part 5

알고리즘

알고리즘

- 연령 계층별 그룹화
 - 연령별로 구분하여 Age Table의 agegroup으로 설정



1	2	3	4	5	6
~18세	19세~23세	24세~28세	29세~33세	34세~39세	40세~

알고리즘

- 분위에 따른 정규화
 - 별점, 총 구매수

```
arr = np.hstack((a1,a2))
result2=[np.quantile(arr,.25),np.quantile(arr,.50),np.quantile(arr,.75)]
print("별점 (1~3 분위):",result2)

a1,a2=arr1,arr2

for jeans in range(len(jean1)):
    if jean1[jeans]['tp'] != '':
        a1[jeans]=jean1[jeans]['tp']
    else:
        a1[jeans]=0

for jeans in range(len(jean2)):
    if jean2[jeans]['tp'] != '':
        a2[jeans]=jean2[jeans]['tp']
    else:
        a2[jeans]=0

arr = np.hstack((a1,a2))
result3=[np.quantile(arr,.25),np.quantile(arr,.50),np.quantile(arr,.75)]
print("총 구매수 (1~3 분위):",result3)
```

- 정규화 할 별점, 총 구매수, 연령별 구매수를
각 분위수에 따라 분류하여 가중치 설정



비정형화된 값을
0~1 사이로 정형화 시키기 위함

- 별점 3분위 가중치

~0.0	0.0~4.7	4.7~4.8	4.8~
0.1	0.2	0.3	0.4

- 총 구매수 3분위 가중치

~0.0	0.0~204.0	204.0~725.0	725.0~
0.1	0.2	0.3	0.4

알고리즘

▪ 분위에 따른 정규화 - 연령별

• 18세 이하 3분위 가중치

~0.0	0.0~14.0	14.0~54.0	54.0~
0.1	0.2	0.3	0.4

• 19세~22세 3분위 가중치

~0.0	0.0~49.0	49.0~198.0	198.0~
0.1	0.2	0.3	0.4

• 23세~27세 3분위 가중치

~0.0	0.0~61.0	61.0~243.5	243.5~
0.1	0.2	0.3	0.4

• 28세~33세 3분위 가중치

~0.0	0.0~29.0	29.0~126.0	126.0~
0.1	0.2	0.3	0.4

• 34세~39세 3분위 가중치

~0.0	0.0~10.0	10.0~44.0	44.0~
0.1	0.2	0.3	0.4

• 40세 이상 3분위 가중치

~0.0	0.0~10.0	10.0~37.5	37.5~
0.1	0.2	0.3	0.4

알고리즘

- 분위에 따른 정규화 (결과 예시)
 - 별점, 총 구매수, 연령 그룹 예시

크롤링

Product_id	Rating	총 구매 수	연령별 구매 수 (6 계층)
858911	4.8	85020	[10000,30000,30000,15000,0,20]
203011	4.6	80	[10,20,20,20,7,3]



Products

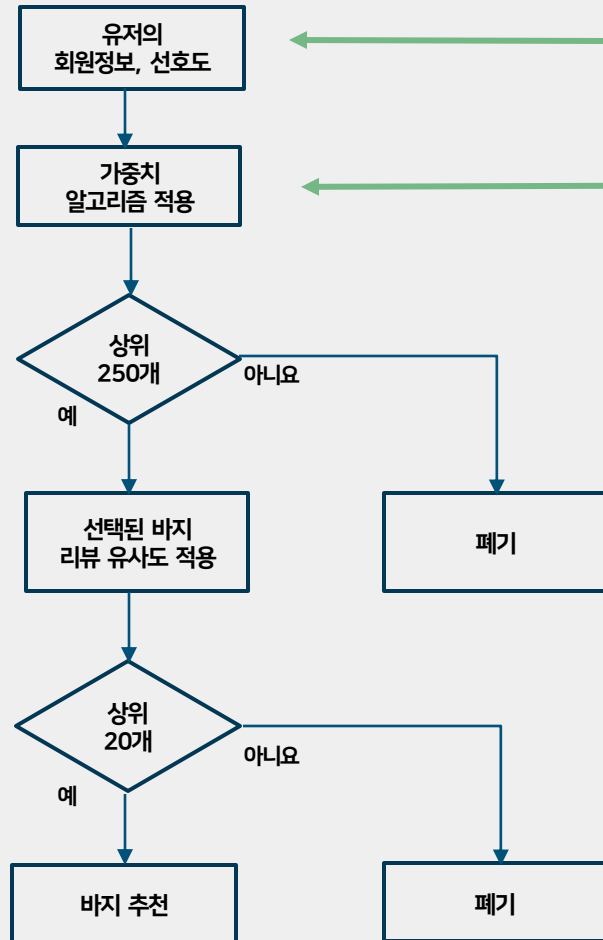
Product_id	Rating	총 구매 수
858911	0.3	0.4
203011	0.2	0.2

Product_age

Product_id	Agegroup	apw
858911	1	0.4
858911	2	0.4
858911	3	0.4
858911	4	0.4
858911	5	0.1
858911	6	0.3

알고리즘

알고리즘 순서도



웹사이트에서 유저의 회원정보와
설문조사 단계에서 선호도 Table 구성

유행에 맞는 옷 : a1

연령에 적합한 옷 : a2

평점이 높은 옷 : a3

Trend (유행의 분위 수): w1

Total Purchase of age(연령별 총구매수의 분위 수) : w2

Rating (별점의 분위 수) : w3

$W = a1 * w1 + a2 * w2 + a3 * w3$

내림차순 결과 상위 250개 상품 번호 선택

선택된 상품에서 리뷰 데이터를 토대로
유저의 회원정보에서 키와 몸무게가 +-5인 데이터를
검색하여 리뷰내 유사도가 높은 상품의 번호를 추출
(선택된 바지의 리뷰에서 나와 신체 사이즈가 있는 값들을
뽑아와 유사도를 측정하여 가장 높은 값을 추출)

$avg(1 - abs((리뷰 키 / 리뷰 몸무게) - (유저 키 / 유저 몸무게)))$
가장 높은 상위 20개 데이터 추출
(코사인 유사도 변형)

추천 결과 웹사이트 내 구현

알고리즘

▪ SQL Join을 활용한 알고리즘

• Product Table

PRODUCT_ID	RATE W	TPW
858911	0.4	0.2
203011	0.3	0.3
191911	0.3	0.1

• Product_Age Table

PRODUCT_ID	AGEGROUP	APW
858911	1	0.4
858911	2	0.3
858911	3	0.3
858911	4	0.2
858911	5	0.1
858911	6	0.4
203011	1	0.4
203011	2	0.2

• Age Table

AGEGROUP
1
2
3
4
5
6

• Preference Table

USERINFO_UI	AGEGROUP	TREND	RATING	PREFER_OF_AGE
A1001	1	0.5	0.3	0.2
A1002	2	0.2	0.3	0.5
A1003	3	0.2	0.5	0.3



Product.tpw * Preference.trend
 Product.ratew * Preference.rating
 Product_age.apw * Preference.prefer_of_age

Sum(~) Order by rownum<=250 order by weight desc

• EX Table (Temporary Table)

PRODUCT_ID	TREND WEIGHT	RATING WEIGHT	PREFER_OF_AGE WEIGHT
858911	2	0.5	0.3
203011	2	0.2	0.3
191911	3	0.2	0.5

알고리즘

▪ SQL Join을 활용한 알고리즘

• Userinfo

USERINFO_UI	AGEGROUP	HEIGHT	WEIGHT
A1001	1	172	69
A1002	2	183	78
A1003	3	179	75

• Preference Table

USERINFO_UI	AGEGROUP	TREND	RATING	PREFER_OF_AGE
A1001	1	0.5	0.3	0.2
A1002	2	0.2	0.3	0.5
A1003	3	0.2	0.5	0.3

• Reviews

PRODUCT_ID	HEIGHT	WEIGHT
858911	170	65
858911	175	70
203011	172	70

• Product_Age Table

PRODUCT_ID	AGEGROUP
858911	1
858911	2
858911	3
858911	4
858911	5
858911	6

• Age Table

AGEGROUP
1
2
3
4
5
6

• Product Table

PRODUCT_ID	RATEW
858911	0.4
203011	0.3
191911	0.3

알고리즘

▪ SQL Join을 활용한 알고리즘

• Reviews

PRODUCT_ID	HEIGHT	WEIGHT
858911	170	65
858911	175	70
203011	172	70

• Product_Age Table

PRODUCT_ID	AGEGROUP
858911	1
858911	2
858911	3
858911	4
858911	5
858911	6

• Product Table

PRODUCT_ID	RATEW
858911	0.4
203011	0.3
191911	0.3

• EX Table (Temporary Table)

PRODUCT_ID	TREND WEIGHT	RATING WEIGHT	PREFER_OF_AGE WEIGHT
858911	2	0.5	0.3
203011	2	0.2	0.3
191911	3	0.2	0.5

‘Product_Age’와 조인관계인 ‘Reviews’는 ‘AgeGroup’에 따라 다르게 조인 하지만, 키와 몸무게는 나이와 연관이 없으므로, 나이에 따라 나뉘져 있는 리뷰를 하나로 묶어서 [EX 테이블]과 조인해야 함

Ex) 사용자의 키와 몸무게가 170, 60 일 경우
 [리뷰 테이블]에서 170 ± 5 , 60 ± 5 인 값을 찾아서 (agegroup)
 ‘Product_id’로 그룹해서 나이에 따라 나뉘져 있는 값을 묶어 줌
 그 후 변형된 코사인 유사도 적용
 유사도가 가장 가까운 ‘product_id(상품 번호)’ 추출
 그 후, [Ex 테이블] 삭제

알고리즘

▪ SQL 최종구문 및 예시

```
create table ex as select distinct(p.id), (p.ratew*pr.pratew + p.tpw*pr.ptpw +
pa.apw*pr.papw) as weight
from products p
join products_age pa
on pa.id= p.id
join age a
on pa.agegroup = a.agegroup
join preference pr
on pr.agegroup = a.agegroup and pr.ui=WWWinputWWW
and rownum<=250 order by weight desc;
select pa.id, avg(1- abs((r.h/r.w)-(u.h/u.w))) as np from userinfo u
join preference pr
on u.ui=pr.ui
join age a
on u.agegroup=a.agegroup
join products_age pa
on pa.agegroup=u.agegroup
join products p
on pa.id=p.id
join reviews r
on pa.id=r.id and r.h-5<=u.h and u.h<=r.h+5 and r.w-5<=u.w and u.w<=r.w+5 and
u.ui= WWWinputWWW
join ex e
on p.id = e.id
group by pa.id;
drop table ex;
```

```
SQL> select pa.id, avg(1- abs((r.h/r.w)-(u.h/u.w))) as np from userinfo u
2 join preference pr
3 on u.ui=pr.ui
4 join age a
5 on u.agegroup=a.agegroup
6 join product_age pa
7 on pa.agegroup=u.agegroup
8 join product p
9 on pa.id=p.id
10 join review r
11 on pa.id=r.id and r.h-5<=u.h and u.h<=r.h+5 and r.w-5<=u.w and u.w<=r.w+5 and u.ui=319
12 join ex e
13 on p.id = e.id
14 group by pa.id;
```

ID	NP
126	1
123	.974559687

```
SQL> drop table ex;_
```

감사합니다!