

제3회 KUIAI 해커톤

패션 인플루언서 추천 플랫폼 구축

| TEAM 드림BIG

경영학과 2018120073 김태영

통계학과 2019150459 구은아

통계학과 2019150446 김혜림

보건정책관리학부 2019250448 김유민



CONTENT

| 제3회 KUIAI 해커톤

| TEAM 드림_BIG

01 개요

- 프로젝트 배경
- 프로젝트 목표
- 추천시스템 흐름도

02 주 기능

: 이미지 기반 유사 스타일 인플루언서 추천

- 데이터 수집 방법
- 모델 학습
- 모델 성능
- 추천 알고리즘 설명

03 부가 기능

: 게시글 텍스트 기반 인플루언서 정보 요약

- 데이터 수집 방법
- 구현 방법

04 결과

- 프로토타입
- 추천시스템 결과
- 한계점 및 발전방향

개요

01,

개요

- 프로젝트 배경
- 프로젝트목표
- 추천 시스템 흐름도

02,

주 기능

- 데이터 수집 방법
- 모델 학습
- 모델 성능
- 추천 알고리즘 설명

03,

부가 기능

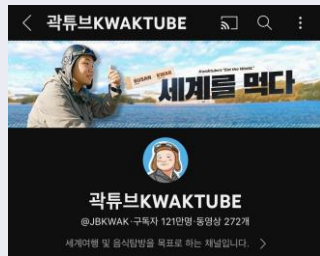
- 데이터 수집 방법
- 구현 방법

04,

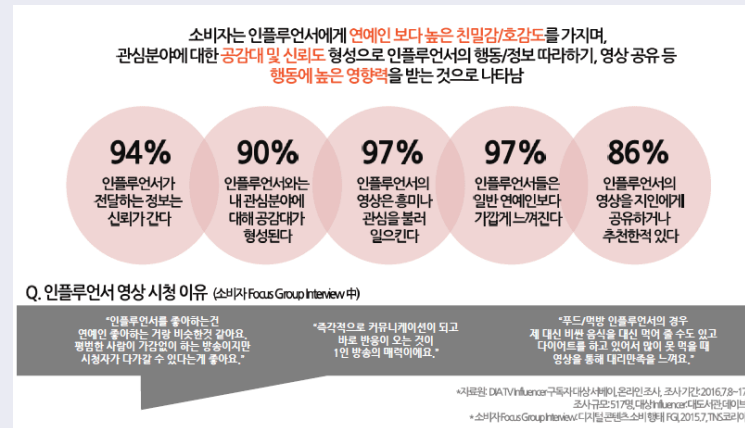
결과

- 프로토타입
- 추천시스템 결과
- 한계점 및 발전 방향

❖ SNS 인플루언서 마케팅 활성화



- **인플루언서** = 많은 팔로워를 보유하며 정보의 전달 및 여론 형성과정에서 영향력을 발휘하는 개인 및 집단
- 실시간으로 소통할 수 있는 각종 온라인과 SNS 상에서 다수의 참여자에게 영향을 끼치고 있음.



- **인플루언서 마케팅**: 기존의 대중적인 스타를 통한 마케팅이 아닌 인플루언서를 통해 브랜드 메시지 전달, 상품판촉, 고객확대를 달성하기 위한 마케팅 전략
- 특히 패션 산업에서 해당 마케팅이 활발히 활용되고 있음
- 인플루언서 자신이 실제 사용한 경험을 바탕으로 SNS에 자신의 일상과 함께 패션 정보를 제공하고 대중들과 자유롭게 소통함

❖ 프로젝트 목표

데이터 기반 패션 인플루언서 추천 알고리즘 구축

POINT 1

패션 상품 정보 및
소비자 정보를 기반으로
추천 시스템 제작

POINT 2

설계된 추천 시스템 기반의
플랫폼 서비스 기획 및
프로토타입 제안

❖ 추천 시스템 사용자

추천 희망 인플루언서 특징

패션 제품 제조사



자사의 제품 홍보 의도에 부합하고
협찬 의뢰 시 가장 그 효과가 클 인플루언서

시스템 상 특징 정의

제품의 스타일과 동일한 스타일로 분류된
피드의 영향력(좋아요 수)이 큰 인플루언서

패션 제품 소비자



소비자의 정보 및 스타일과 가장 유사해 패션 연출 및
코디 정보를 파악할 수 있는 인플루언서

입력하는 패션 사진의
스타일의 피드가 많은 인플루언서

❖ 예상 추천 결과



주 기능 : 이미지 기반 유사 스타일 인프루언서 추천

01,

개요

- 프로젝트 배경
- 추천 시스템 활용 목표
- 추천 시스템 흐름도

02,

주 기능

- 데이터 수집 방법
- 모델 학습
- 모델 성능
- 추천 알고리즘 설명

03,

부가 기능

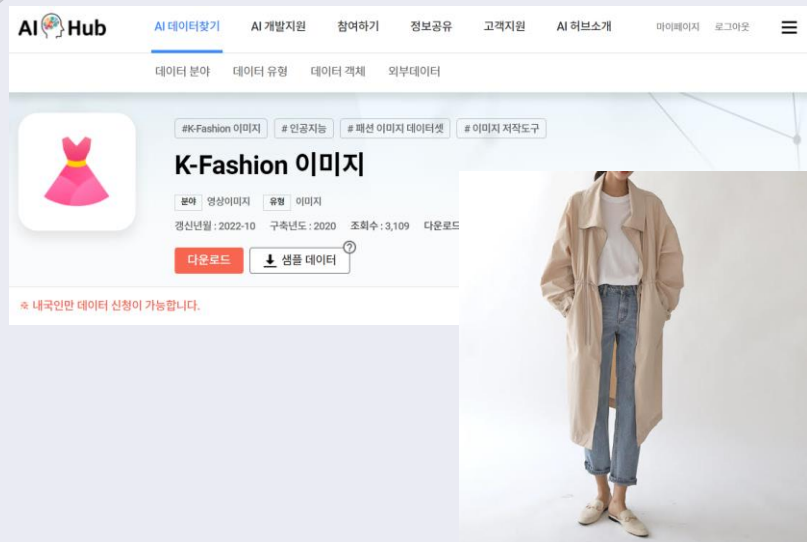
- 데이터 수집 방법
- 구현 방법

04,

결과

- 프로토타입
- 추천시스템 결과
- 한계점 및 발전 방향

❖ 데이터 수집 – 패션 이미지



- AI허브에서 120만건 분량의 패션 이미지 사진을 공개.
- 이미지에서 패션 영역과 속성, 스타일 정보를 인식 및 도출 수 있도록 레이블링을 구성.

트래디셔널	매니시	페미닌	에스닉	컨템포러리	내추럴	젠더플루이드	스포츠	서브컬처	캐주얼
클래식	매니시	페미닌	히피	모던	컨트리	젠더리스	스포츠	레트로	밀리터리
프레피	돌보이	로맨틱	웨스턴	소피스트 케이티드	리조트			키치/ 키덜트	스트리트
		섹시	오리엔탈	아방가르드				힙합	
								펑크	

```

+전문가 레이블링 결과
{ "파일번호": 1, "
파일이름": "REIGN_001_04.jpg", //파일명
"메트차표": { "아우터": { {} }, //좌측 X,Y좌표 / 우측 X,Y좌표
"하의": { {} },
"원피스": { {} },
"상의": { {} },
"폴리곤좌표": { "아우터": { {} }, //좌측 X,Y좌표 / 우측 X,Y좌표
"하의": { {} },
"원피스": { {} },
"상의": { {} },
"라벨링": {
{"스타일": { "스타일": "스트리트" },
"아우터": { { "가장": "롱",
"카테고리": "점퍼", //분류항목
"디테일": { "스트링", "자퍼",
"프린트": { "무지",
"핏": "오버사이즈" },
"하의": { { "가장": "발목",
"카테고리": "청바지",
"디테일": { "롤업",
"소재": { "데님",
"핏": "노멀" },
"원피스": { {} },
"상의": {
{ { "카테고리": "티셔츠",
"소재": { "지지",
"프린트": { "무지",
"넥라인": "라운드네",
"핏": "루즈" } } }
}
  
```

- 위의 표와 같이 스타일 23가지와 세부속성 186가지로 라벨링된 이미지 데이터
- 용량 및 컴퓨팅 자원의 한계로 제공된 데이터셋 중 72,452장을 스타일 클래스 별 골고루 샘플링해 학습에 사용함
- 하나의 이미지는 800x800 사이즈로 jpg와 json 파일 형태로 제공됨

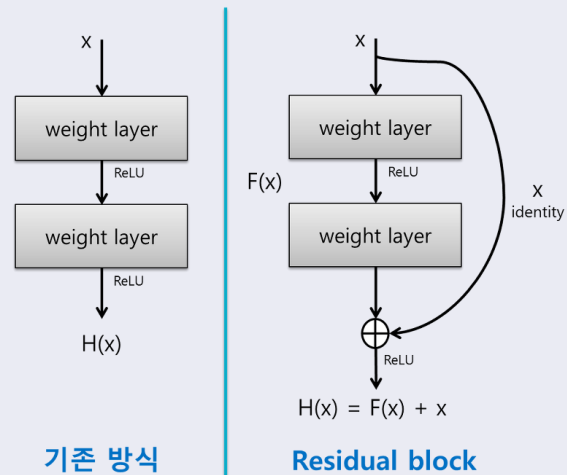
❖ 데이터 수집 - 인스타그램 좋아요 수

```
def get_num_like(driver):  
    try:  
        html = driver.page_source  
        soup = BeautifulSoup(html, "lxml")  
        like = soup.select("div._aaci._aacw._aacw._aada._aade")[1].findAll('span')[-1].text  
    except:  
        like = 0  
    return like  
  
def move_next(driver):  
    try:  
        right = driver.find_element(By.CSS_SELECTOR, "div._aag._aah")  
        right.click()  
        time.sleep(3)  
    except:  
        print("finished")
```



- 피드별 좋아요 수 크롤링을 위해 BeautifulSoup와 Selenium 패키지를 사용
- 인스타그램 로그인부터 계정 검색 진입까지 자동화하여 크롤링을 진행
- 좋아요 수를 비공개 한 피드의 경우 동일 계정의 다른 피드들의 좋아요 수 의 평균값으로 대체
- 크롤링 후 해당 피드와 매칭하여 데이터 저장

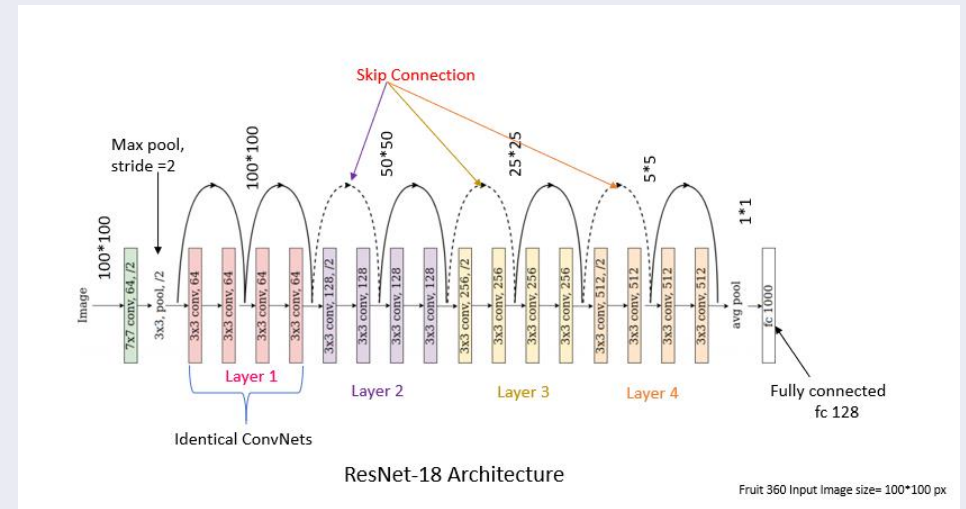
❖ 모델 선정 – ResNet



- 이전에서 설명한 데이터를 활용해 23가지 스타일 클래스를 분류하는 모델로 ResNet을 선정 (이 외 모델 시도 내용은 후술)
- Residual Net은 고안 이전의 모델들이 가졌던 고질적인 vanishing gradient 문제를 해결했다는 평가를 받으며 image classification태스크에서 자주 사용됨
- 가장 큰 특징으로 한 layer의 input이 순차적으로 다음 layer로 전달되는 것이 아닌 input 값 그대로 다른 layer로 전달해 출력값에 더해주는 **Residual Block**

이

존재함



- 구체적으로 분류를 위해 사용한 모델은 18개의 layer의 ResNet을 사전학습한 **파이토치의 pretrained resnet-18**임
- ResNet-18은 위 사진과 같은 구조를 가짐

❖ 모델 학습 기타 설정

Resource

Google Colab GPU

Train / Test split

- **Train** : 90% (65,206장)
- **Test** : 10% (7,246장)

Result Save

학습된 모델
Pytorch pt 파일 저장

Preprocessing

- **Random Resize Crop** : random하게 crop 후 지정한 크기만큼 resize
- **Random Horizontal Flip** : 이미지의 좌우 반전
- **Normalize** : 각 채널 별 이미지 픽셀 값을 채널 별 평균으로 뺀 후 표준편차로 나누는 정규화

Setting

- **Learning Epoch** = 10
- **Loss Criterion** = Cross Entropy Loss
- **Optimizer** = SGD (learning rate = 0.001 / momentum = 0.9)
- **Learning Scheduler** = StepLR (step size = 7 / gamma = 0.1)

❖ 모델 성능 및 기타 시도

Epoch 1/10	Epoch 6/10
Train Acc : 0.3264	Train Acc : 0.4111
Val Acc : 0.3598	Val Acc : 0.3995
Epoch 2/10	Epoch 7/10
Train Acc : 0.3659	Train Acc : 0.4168
Val Acc : 0.3805	Val Acc : 0.4045
Epoch 3/10	Epoch 8/10
Train Acc : 0.3806	Train Acc : 0.4343
Val Acc : 0.3799	Val Acc : 0.4104
Epoch 4/10	Epoch 9/10
Train Acc : 0.3944	Train Acc : 0.4337
Val Acc : 0.3937	Val Acc : 0.4075
Epoch 5/10	Epoch 10/10
Train Acc : 0.4024	Train Acc : 0.4428
Val Acc : 0.3948	Val Acc : 0.4133

최종 성능 = 0.4133
(test accuracy 기준)

- 이는 모델이 23가지 class를 분류하는 것을 염두했을 때 **분류모델**이 좋은 성능을 보인다고 판단함
- 23가지 class를 단순 랜덤 분류할 경우 4.35% 정도의 정확도를 보이는 것에 비해 **10배 가량 더 좋은 성능을 냄**
- 해당 모델로 이후 추천시스템을 구축

기타 시도 내용

(1) 이미지 전처리

- Random Crop : 150 ~ 250 범위 내에서 조정
- Color Jitter : brightness, contrast 조정

(2) 시도 모델 (모델 별 하이퍼파라미터 조정)

- ViT
- ResNet-50
- ResNet-101
- EfficientNet

(3) 기타

- Optimizer : momentum(0.3, 0.5, 0.7, 0.9) / learning rate(0.001, 0.005, 0.01)
- Learning Rate Scheduler : LambdaLR, MultistepLR

❖ 추천 알고리즘 내용 - 제조사

Step1. 입력이미지에 대한 스타일 분류



- 왼쪽과 같은 상품 이미지를 입력할 경우, 상품 자체에 대한 스타일을 분류하고 이에 대한 인플루언서를 추천
- 오른쪽과 같은 컨셉 이미지를 입력할 경우, 인플루언서가 상품을 착용했을 때 바라는 전체적인 스타일을 분류하고 이에 대한 인플루언서를 추천
- 위 과정은 이전에서 설명한 스타일 클래스 분류 단일 모델로 구현 가능

Step2. 인플루언서 피드 스타일 분류



	user_name	post_id	text	timestamp	like	video	post_style_idx	post_style
0	Onefence	3008175603962386523	어제 업로드 하려고 했어요 진짜예요	2023-01-04 8:54:58	526	0	0	기타
1	Onefence	3006778853318091854	옷장에 옷은 많은데 당장 내일은 뭐 입어야 될지 모르겠어요...?? 🤔	2023-01-02 10:39:52	-1	0	10	스포츠
2	Onefence	3006161204775157958	#협찬 @myflpr_official 윈윈다들 새해 첫날 어떻게 보내셨나요? 윈...	2023-01-01 14:12:43	521	0	0	기타
3	Onefence	3005347722609627475	마지막 인사 올립니다 꾸벅	2022-12-31 11:16:28	532	0	9	스트리트
4	Onefence	3004577405444400836	한살 먹는 건 모르겠고 빨리 쉬고싶어요 ㅁ	2022-12-30 9:45:59	-1	0	0	기타

- 인플루언서의 각 게시글의 사진을 학습 및 저장한 분류 모델로 스타일을 분류
- 모든 피드의 이미지에 대해 분류를 수행
- 게시글에 이미지가 없는 경우에는 no image 클래스로 분류
- 위의 by_he.nique 인플루언서 사진에 대해 페미닌으로 모델이 분류하고 분류 결과를 피드별로 저장

❖ 추천 알고리즘 내용 - 제조사

Step3. (제조사) 인플루언서 스타일 테이블 구축

	user_name	post_id	like	post_style		post_style	like
0	0nefence	3008175603962386523	526	기타	0	스트리트	4286
1	0nefence	3006778853318091854	-1	스포티	1	페미닌	1538
2	0nefence	3006161204775157958	521	기타	2	기타	1517
3	0nefence	3005347722609627475	532	스트리트	3	스포티	1277
4	0nefence	3004577405444400836	-1	기타	4	모던	1243
5	0nefence	3003173699142298173	547	밀리터리	5	밀리터리	1225
6	0nefence	3002428195739727239	534	페미닌	6	리조트	554

- 제조사의 추천시스템 활용 목적은 인플루언서를 통한 홍보이기 때문에 인플루언서의 영향력을 중시할 것으로 가정
- 따라서 인플루언서의 피드에 등장하는 스타일 중 좋아요 개수가 가장 많은 피드의 스타일을 인플루언서의 대표 스타일로 선정
- 위 계정의 경우 스트리트 스타일로 분류된 피드의 좋아요 수가 가장 많았으므로 최종 스트리트로 결정

Step4. 성별 및 사이즈 반영 후 최종 추천



- Step3의 제조사 인플루언서 스타일 테이블을 사용
- 제조사가 입력한 이미지의 스타일 분류 결과, 성별, 사이즈를 사용해 필터링
- 필터링 이후 좋아요 개수로 sort하여 상위 3명의 인플루언서의 아이디를 반환
- 이 때, 피드의 일련번호도 같이 저장했기에 추천하는 인플루언서의 피드 사진도 결과창에 반환

❖ 추천 알고리즘 내용 - 소비자

Step1. 입력이미지에 대한 스타일 분류



- 왼쪽과 같은 본인의 패션 이미지를 입력할 경우, 본인이 평소 입고 다니는 스타일을 가진 인플루언서를 추천
- 오른쪽과 같은 본인 취향의 코디 이미지를 입력할 경우, 본인이 원하는 스타일을 가진 인플루언스를 추천
- 위 과정은 이전에서 설명한 스타일 클래스 분류 단일 모델로 구현 가능.

Step2. 인플루언서 피드 스타일 분류 (제조사와 동일)



	user_name	post_id	text	timestamp	like	video	post_style_idx	post_style
0	Onefence	3008175603962386523	어제 업로드 하려고 했어요 진짜예요	2023-01-04 8:54:58	526	0	0	기타
1	Onefence	3006778853318091854	옷장에 옷은 많은데 당장 내일은 뭐 입어야 될지 모르겠어요...?? 🤔	2023-01-02 10:39:52	-1	0	10	스포츠
2	Onefence	3006161204775157958	#협찬 @myflpr_official 윈윈다들 새해 첫날 어떻게 보내셨나요? 윈...	2023-01-01 14:12:43	521	0	0	기타
3	Onefence	3005347722609627475	마지막 인사 올립니다 꾸벅	2022-12-31 11:16:28	532	0	9	스트리트
4	Onefence	3004577405444400836	한살 먹는 건 모르겠고 빨리 쉬고싶어요 ㅁ	2022-12-30 9:45:59	-1	0	0	기타

- 인플루언서의 각 게시글의 사진을 학습 및 저장한 분류 모델로 스타일을 분류
- 모든 피드의 이미지에 대해 분류를 수행
- 게시글에 이미지가 없는 경우에는 no image 클래스로 분류
- 위의 by_he.nique 인플루언서 사진에 대해 페미닌으로 모델이 분류하고 분류 결과를 피드별로 저장

❖ 추천 알고리즘 내용 - 소비자

Step3. (소비자) 인플루언서 스타일 테이블 구축

	user_name	post_id	like	post_style	post_style	like
2089	ruri.kim	3008664416573763299	-1	페미닌	0 페미닌	14
2090	ruri.kim	3008264260039455475	-1	리조트	1 모던	6
2091	ruri.kim	3007910140960418013	-1	페미닌	2 스트리트	3
2092	ruri.kim	3007215101049418953	-1	스트리트	3 로맨틱	2
2093	ruri.kim	3006464752177115993	-1	모던	4 클래식	2
2094	ruri.kim	3005889960149857521	-1	페미닌	5 기타	1
					6 리조트	1
					7 매니시	1

- 소비자의 추천시스템 활용 목적은 인플루언서를 통한 패션 정보 탐색 및 획득이기에 인플루언서 피드의 빈도를 중시할 것으로 가정
- 위 계정의 경우 페미닌 스타일로 분류된 피드의 수가 가장 많아 최종 페미닌으로 결정

Step4. 성별 및 사이즈 반영 후 최종 추천



- Step3의 소비자 인플루언서 스타일 테이블을 사용.
- 소비자가 입력한 이미지의 스타일 분류 결과, 성별, 사이즈를 사용해 필터링
- 필터링 이후 좋아요 개수로 sort하여 상위 3명의 인플루언서의 아이디를 반환
- 이 때, 피드의 일련번호도 같이 저장했기에 추천하는 인플루언서의 피드 사진도 결과창에 반환

부가 기능 : : 게시글 텍스트 기반 인플루언서 정보 요약

01,

개요

- 프로젝트 배경
- 프로젝트 목표
- 추천 시스템 흐름도

02,

주 기능

- 데이터 수집 방법
- 모델 학습
- 모델 성능
- 추천 알고리즘 설명

03,

부가 기능

- 데이터 수집 방법
- 구현 방법

04,

결과

- 프로토타입
- 추천시스템 결과
- 한계점 및 발전 방향

❖ 부가 기능 개요

- 단순 계정만을 반환할 시 **추천 결과에 대한 근거를 제시하지 못하고** 이용자가 다시 인플루언서에 대해 탐색하는 추가적인 **비효율이 발생함**
- 인플루언서에 대한 정보를 **직관적으로** 제시하고자 함
- 이를 위해 각 인플루언서별 정보를 요약할 수 있는 **워드클라우드**를 생성함

❖ 분석 과정

STEP 1

데이터 통합 및 불용어 처리

각 유저별 피드를 하나로 통합
불용어 사전 기반
인스타 한정 불용어 정의

어제는 야근은 안 하고 회식을 .. 🙄 흑식
야근 vs 회식 뭐가 더 싫으세요 (?) 📺 📺
🙄 피곤해서 대충 찍은,,, 크흑,,, 자철,,
이야,,, 앓게 해조라 😞

STEP 2

토큰화

Mecab 활용하여 진행

STEP 3

명사별 빈도수 확인 및 워드 클라우드 생성

명사별 counter 객체 사용
일반 명사 및 고유 명사 추출

어제 는 야근 은 안 하 고 회식 을 흑 식 야
근 회식 뭐 가 더 싫 으세요 피곤 해서 대충
찍 은 크 흑 자 철 이 야 앓 게 해 조 라

결과

01,

개요

- 프로젝트 배경
- 프로젝트 목표
- 추천 시스템 흐름도

02,

주 기능

- 데이터 수집 방법
- 모델 학습
- 모델 성능
- 추천 알고리즘 설명

03,

부가 기능

- 데이터 수집 방법
- 구현 방법

04,

결과

- 프로토타입
- 추천시스템 결과
- 한계점 및 발전 방향

❖ 프로토타입

- PyQt5 라이브러리를 활용하여 구현함
- 여타 라이브러리에 비해 손쉽게 GUI를 구현할 수 있음
- 구현하고자 하는 서비스에 가장 적합한 인터페이스를 제공 가능

POINT 1

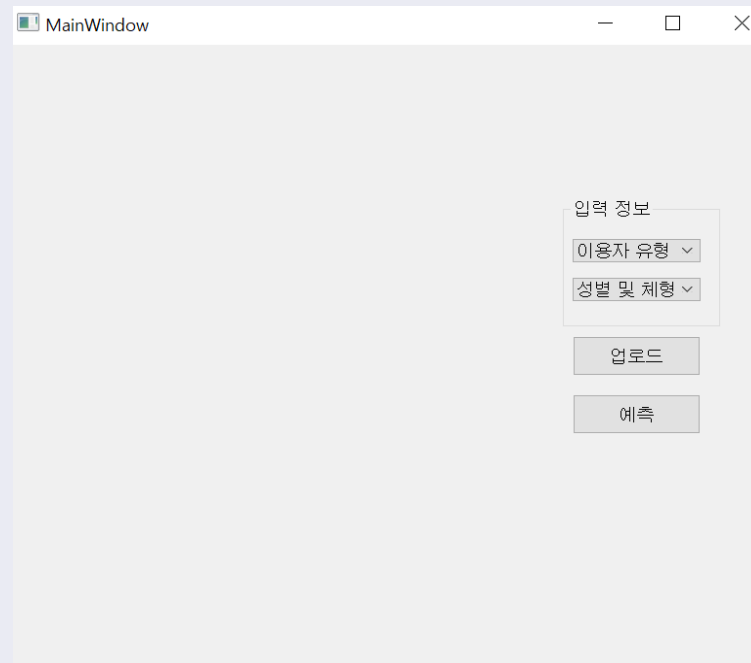
실행 후 첫 화면

- 이용자 유형(제조자, 소비자)
- 인플루언서 성별 및 체형
(여S,M,L 남S,M,L)

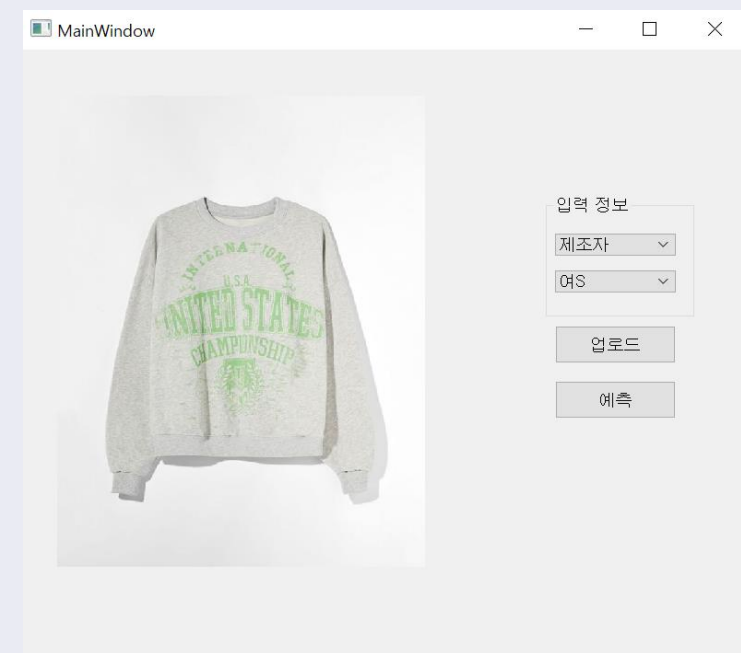
옵션을 입력 후 이미지 업로드

*업로드 하지 않은 경우 예측 버튼 비활성화

입력전

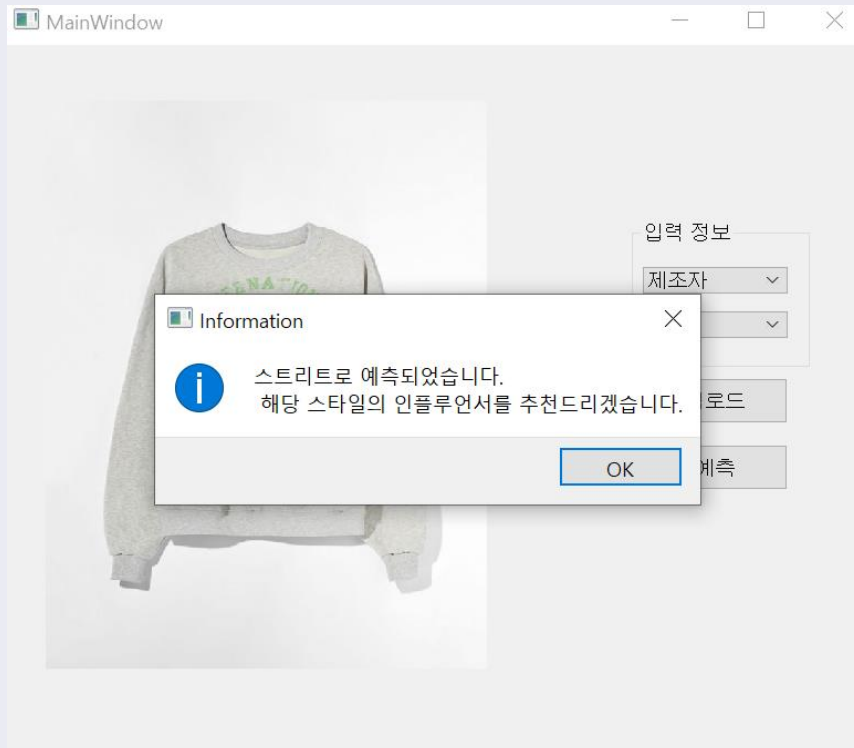


입력후



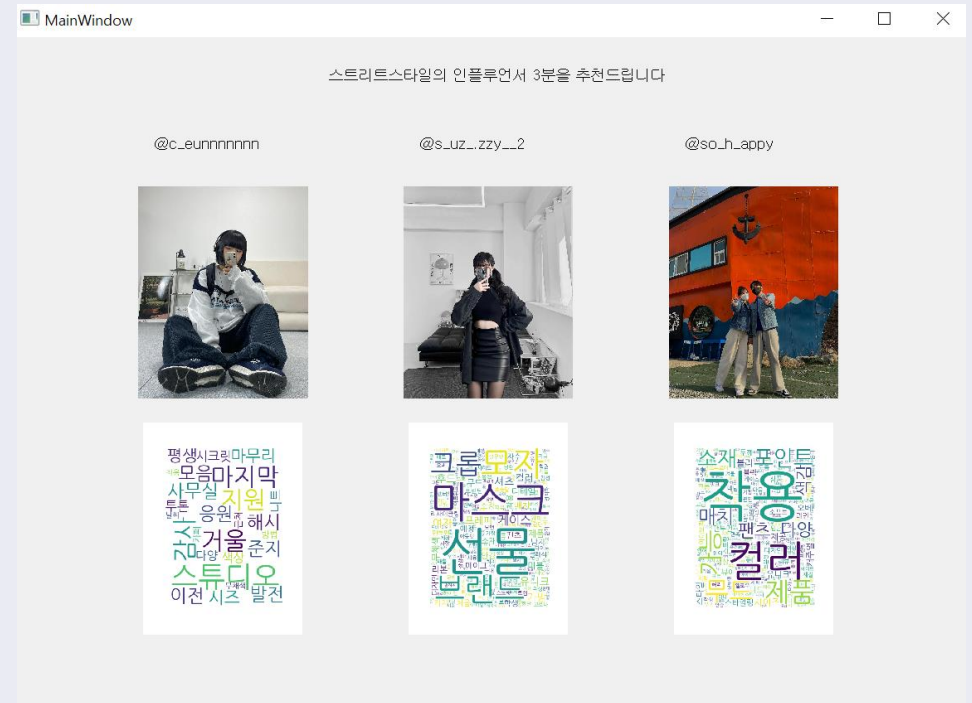
❖ 프로토타입

예측화면 1



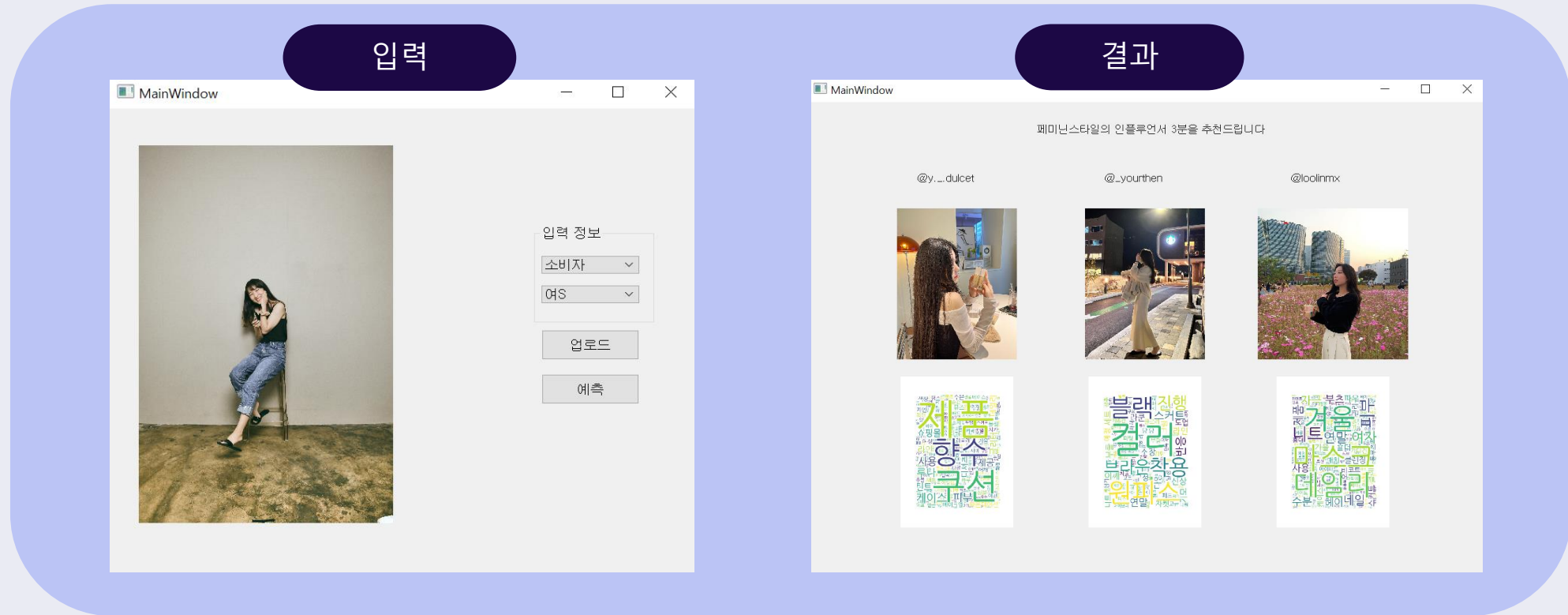
예측 버튼 누르면
스타일 예측 결과창이 활성화됨

예측화면 2



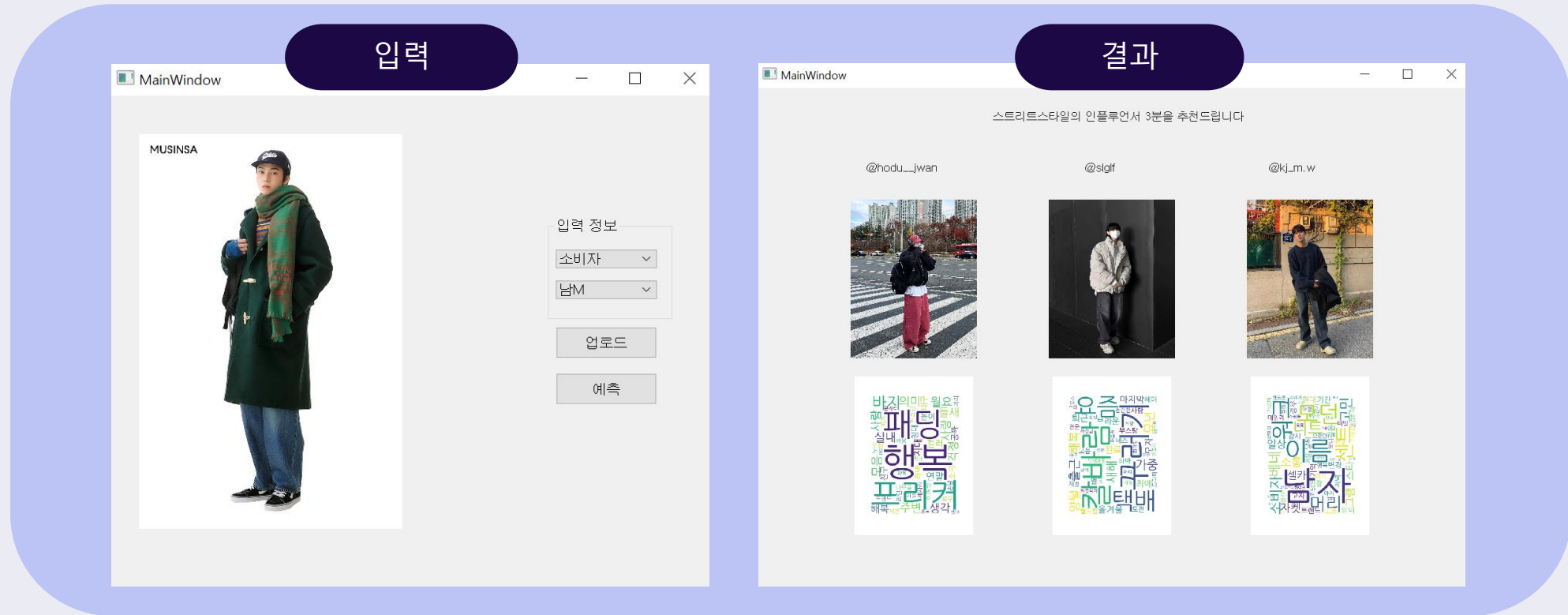
새로운 창에서 추천 시스템 기반으로
추천된 3명의 인플루언서의
아이디, 대표 피드, 워드클라우드 정보 제공

❖ 프로토타입 결과 – 소비자 (1)



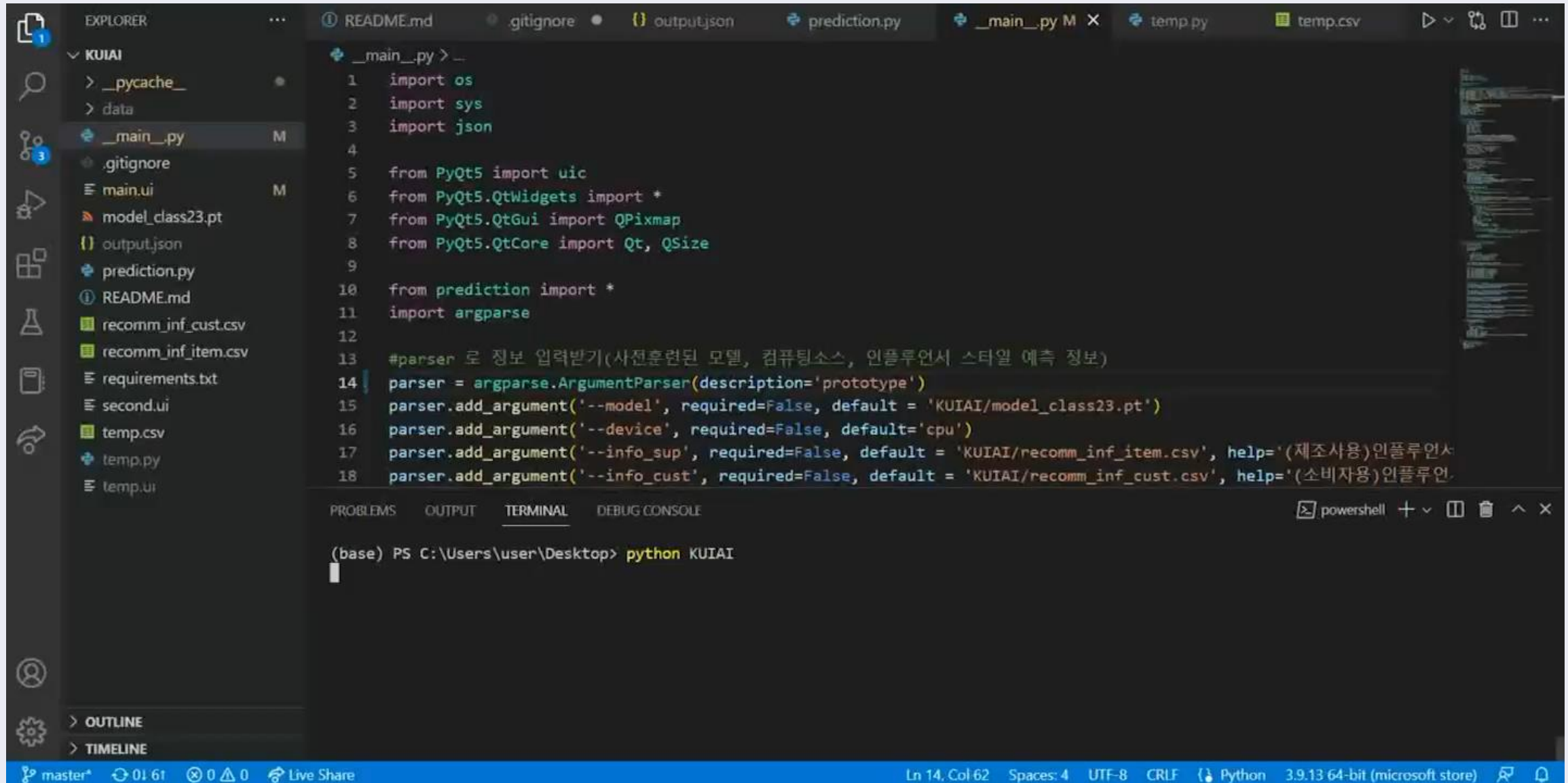
- 일반 이용자 사진 입력 후 적절한 옵션을 선택 후 예측 진행
- 해당 사진의 스타일을 페미니인으로 예측하며 이에 따라 세 명의 인플루언서를 추천함
- 제공되는 인플루언서의 대표 피드의 스타일을 확인했을 때, 본 프로토타입은 합리적인 결과를 제공하고 있음

❖ 프로토타입 결과 – 소비자 (2)



- 해당 사진의 스타일을 스트리트로 예측하며 이에 따라 세 명의 인플루언서를 추천함
- 앞선 결과와 마찬가지로 합리적인 결과 확인 가능

❖ 추천 시스템 시연



The screenshot displays the Visual Studio Code interface for a project named KUIAI. The Explorer panel on the left shows the project structure, including files like `__main__.py`, `main.ui`, `model_class23.pt`, `output.json`, `prediction.py`, `README.md`, `recomm_inf_cust.csv`, `recomm_inf_item.csv`, `requirements.txt`, `second.ui`, `temp.csv`, `temp.py`, and `temp.ui`. The `__main__.py` file is open in the editor, showing the following code:

```
1 import os
2 import sys
3 import json
4
5 from PyQt5 import uic
6 from PyQt5.QtWidgets import *
7 from PyQt5.QtGui import QPixmap
8 from PyQt5.QtCore import Qt, QSize
9
10 from prediction import *
11 import argparse
12
13 #parser 로 정보 입력받기(사전훈련된 모델, 컴퓨팅소스, 인플루언서 스타일 예측 정보)
14 parser = argparse.ArgumentParser(description='prototype')
15 parser.add_argument('--model', required=False, default = 'KUIAI/model_class23.pt')
16 parser.add_argument('--device', required=False, default='cpu')
17 parser.add_argument('--info_sup', required=False, default = 'KUIAI/recomm_inf_item.csv', help='(제조사용)인플루언서')
18 parser.add_argument('--info_cust', required=False, default = 'KUIAI/recomm_inf_cust.csv', help='(소비자용)인플루언서')
```

The terminal at the bottom shows the command `python KUIAI` being executed in a PowerShell window.

❖ 한계점 및 발전방향

한계점

- **인플루언서 선정 방식**
인플루언서는 해시태그로 특정 스타일을 언급하더라도 해당 스타일의 피드를 게시하지 않음. 따라서 해시태그를 기반으로 크롤링된 인플루언서를 이용하여 이용자에게 추천하는 방식은 다양한 스타일을 추천할 수 없다는 점에 한계가 있음.
- **다양한 체형의 인플루언서 풀 확보 x**
이용자의 체형에 맞는 인플루언서를 추천해주기 위해서는 그만큼 다양한 체형의 인플루언서가 필요함. 하지만 현재 여L, 남S 체형의 인플루언서 수가 타 체형에 비해서 적음.
- **데이터 퀄리티**
현재 스타일 간 경계가 뚜렷하지 않은 이미지가 다수 존재함.
이는 스타일 분류 모델을 학습할 때에 영향을 줌. 따라서 정확하게 스타일이 분류된 이미지 셋을 활용하여 모델의 성능을 높일 수 있을 것.

발전 방향

- **체형의 객관적 기준 수립**
- **상품정보 예측 모델링**
상품 정보까지 예측이 가능한 모델을 구축한다면 이용자가 입력한 것과 동일한 카테고리의 상품을 추천할 수 있음 이를 통해 구체적으로 입력한 상품을 어떻게 활용할 것인지 이용자에게 제안이 가능함.

감사합니다

01,

개요

- 프로젝트 배경
- 프로젝트 목표
- 추천 시스템 흐름도

02,

주 기능

- 데이터 수집 방법
- 모델 학습
- 모델 성능
- 추천 알고리즘 설명

03,

부가 기능

- 데이터 수집 방법
- 구현 방법

04,

결과

- 프로토타입
- 추천시스템 결과
- 한계점 및 발전방향

Reference

Lee, W. T., Cha, M., & Yang, H. R. (2011). 소셜미디어 유력 자의 네트워크 특성: 한국의 트위터 공동체를 중심으로 [Network properties of social media influentials: Focusing on the Korean Twitter community]. Journal of Communication Research, 48(2), 44–79.

Choi, J., & Kang, H. K. (2017, September 15). 소셜 인플루언 서를 활용한 미국시장 진출 전략, Global Market Report17-033 [Strategies to advance into the US market using social influencer, Global Market Report 17-033]. KOTRA. Retrieved October 30, 2017.

김우빈, 추호정(2018) Dept. of Textiles, Merchandising and Fashion Design, Seoul National University *Research Institute of Human Ecology, Seoul National University Received May 28, 2018; Revised (October 5, 2018; November 14, 2018); Accepted November 30, 2018.

손동진, 김혜경. (2017). 소셜 인플루언서를 활용한 디지털 마케팅 전략 연구: 올레드 TV 글로벌 디지털 캠페인 사례를 중심으로. 광고 PR 실학연구, 10(2), 64-95.

최지윤, 이규혜. (2022). 패션 추천 서비스 알고리즘에서 상품유형과 속성 조합의 영향.

이재영, 김영훈. (2021). 패션사진의 좋아요를 이용한 인스타그램 사용자 임베딩과 패션 사진 추천.

김건우. (2018). 영상처리기술을 이용한 사용자 맞춤 패션 추천 어플리케이션.

이정호, 강필성. (2021). 그래프와 이미지 기반의 패션 아이템 추천.

Muhammad Omer. (2022). 딥러닝 기반 개인화 패션 추천 시스템.

파이토치 공식 홈페이지 - ResNet (https://pytorch.org/hub/pytorch_vision_resnet/)

Kaiming He, Xiangyu Zhang. (2015). Deep Residual Learning for Image Classification.

Kaiming He, Xiangyu Zhang. (2016). Identity Mappings in Deep Residual Networks.