

# 구름구름

내옷장기반스타일링추천서비스  
Closet Cloud  
2조



# 목차 a table of contents

---

- 1 프로젝트 개요
- 2 프로세싱
- 3 프로젝트 결과 및 기대효과
- 4 개발 후기 및 QnA

# 프로젝트 개요

---

1. 프로젝트 기획 배경
2. 구성원 및 역할
3. 프로젝트 일정



# 프로젝트 기획 배경

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud

이번 주 토요일에 결혼식 있는데 옷 뭐 입고 가지?



이 옷,,, 내가 이런 옷이 있었나?

이번 소개팅 때 뭐 입고 나가지?

어떤 옷을 매칭하는게 좋을 까?

아 옷은 많은데 입을게 없네



AI Styling Recommendation Service  
Closet Cloud

이제 봄인데 무슨 옷을 사지?



어떤 옷을 사야할지 모르겠다.

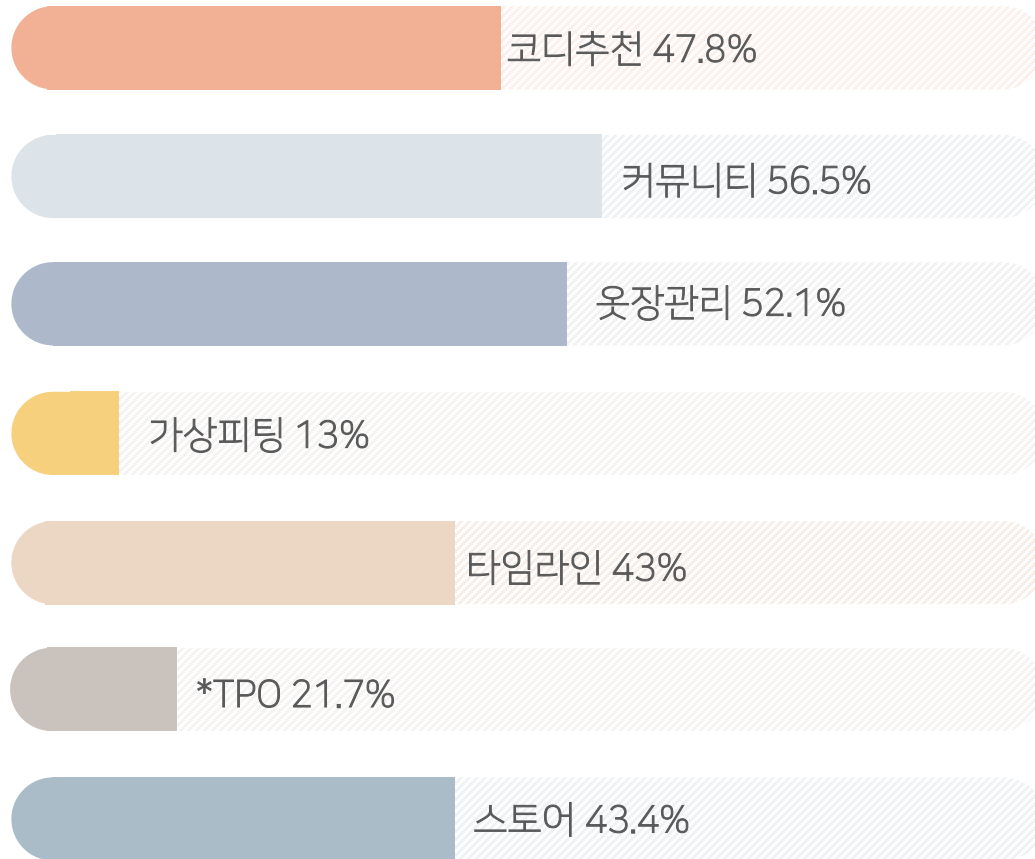
옷이 없다 없어

어떤 옷을 구매하는 게 좋을까?

# 프로젝트 기획 배경

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud

< 앱스토어 내의 어플 별 기능 조사 [총 23개]>



\* TPO: TIME(시간), PLACE(장소), OCCASION(상황)에 따라 의복을 알맞게 착용하는 것

## 조사 결과

- 가상피팅의 경우 실제 이미지와 보여지는게 다름
- TPO는 실용성이 부족함
- 코디 추천은 유저간 추천이 대부분
- 커뮤니티나 타임라인 이용 활발
- AI코디 제공 어플은 단 1곳

## 추가

- 옷의 선호도를 체크하여 즐겨찾기 기능 유용
- 옷의 착용 횟수를 기록하여 관리하는 기능
- 리더 보드를 활용하여 유저들의 정보 제공량 상승

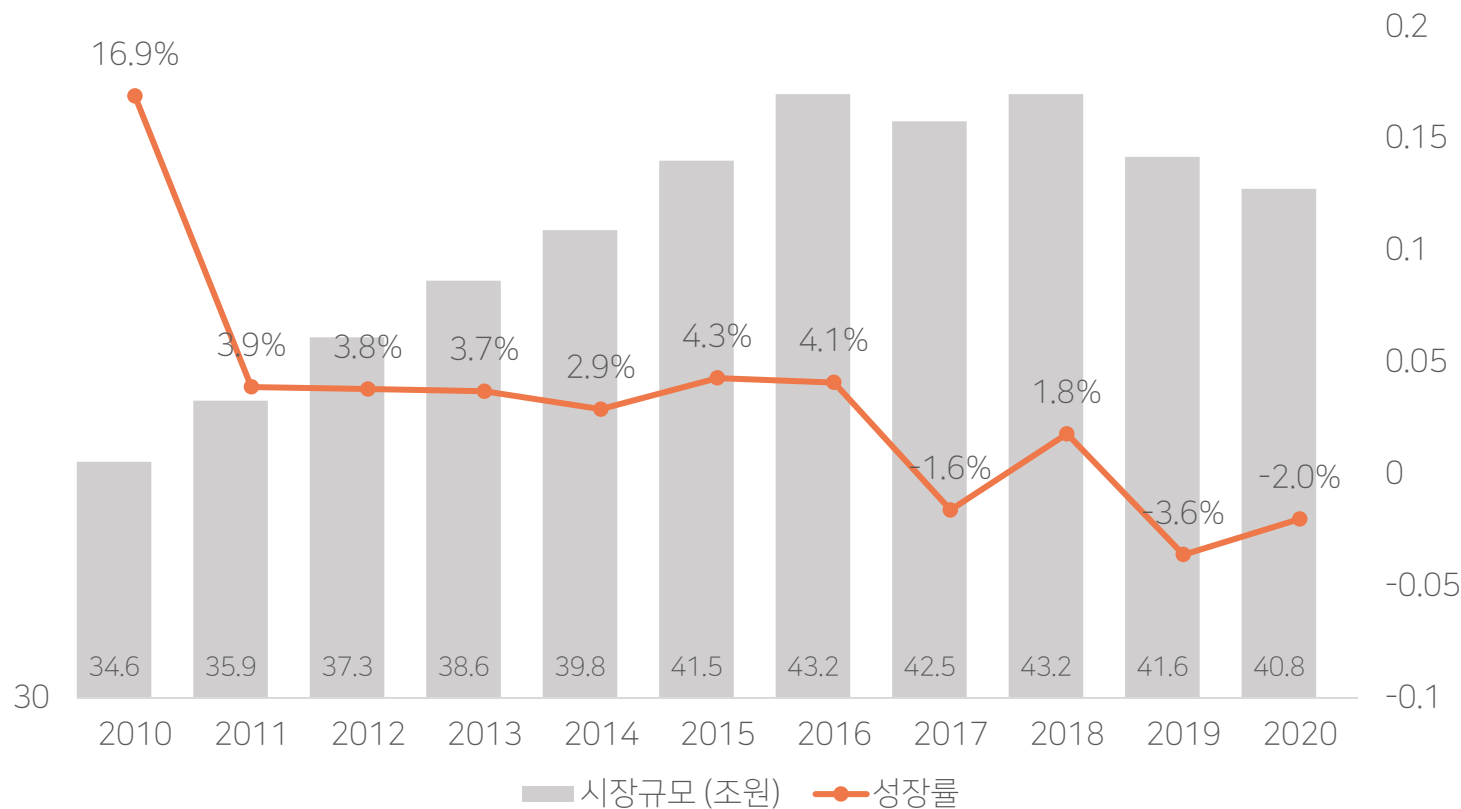
## 결론

이용하기 **부족한 부분들을 개선**하고,  
좋은 기능을 함께 구현하여  
**보다 나은 서비스를 제공** 가능

# 프로젝트 기획 배경

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud

한국 패션시장 규모 및 성장률



## 분석

- 코로나19로 인하여 성장률 저조
- 2020년 이후 점차 회복세 전망
- 거리두기완화로 인한 소비량 증가

## 시장 증감률

- 여성정장시장 규모 17.5% 감소
- 남성정장시장 규모 9.9% 감소
- 아동복시장 규모 22.4% 감소
- 스포츠복시장 규모 7.6% 상승
- 내의시장 규모 3.5% 상승
- 캐주얼복시장 규모 1.9% 상승
- 신발시장 규모 5.7% 상승
- 가방시장 규모 8.8% 상승

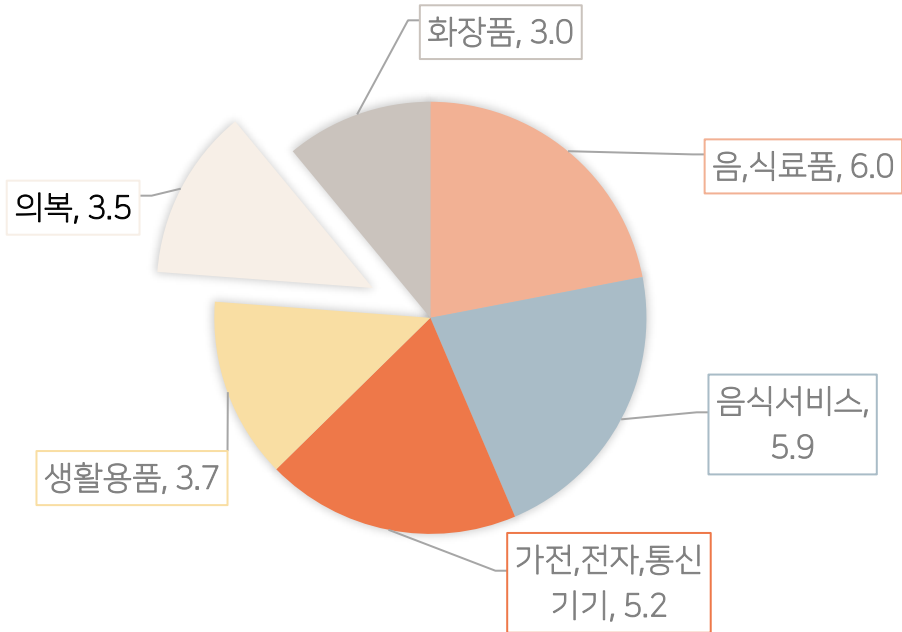
<자료 -한국섬유산업연합회 '한국 패션마켓 트렌드' 보고서>



# 프로젝트 기획 배경

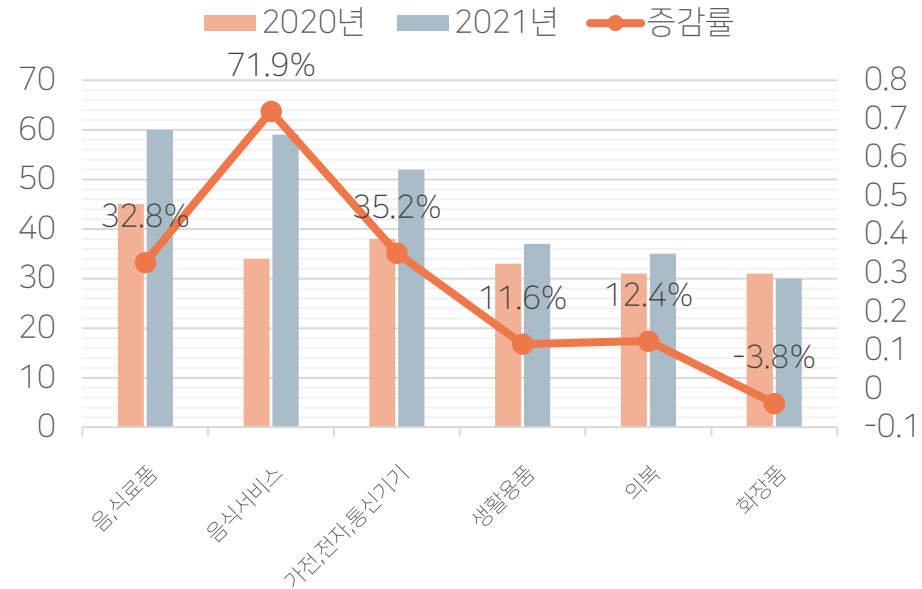
빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud

21년 온라인 쇼핑 거래액 추이(단위: 조원)



<자료 -통계청>

항목별 증감률 (단위: 백만원, %)



## 조사 결과

- 21년 온라인 쇼핑 거래액은 총 44조 7천억원
- 의복은 35조원으로 총 거래액 기준 적은 수준
- 작년대비 증감률 124%, 코로나19 여파가 적용
- 음식 서비스의 경우, 서비스 경쟁 작용으로 인한 성장 극대화
- 패션시장에서도 질 높은 서비스 필요

# 구성원 및 역할



팀장 이택형

 [github newkaist](#)

- 전반적인 프로젝트 관리 (일정, 수행방안 등)
- 데이터 전처리 및 라벨링
- 스타일 추천 시스템 구성 및 테스트
- 모델 성능 개선
- 서버 구현



부팀장 이지현

 [github jibook](#)

- 팀장 부재 시, 팀장 역할 대리 수행
- 진행 내역 정리 및 보고 자료 작성
- 분류 모델 (VGG16, ResNet101) 구성 및 테스트
- 모델 성능 개선
- 이미지 입력 구축
- 서비스 화면 기능 구현



팀원 손지원

 [github sonzwon](#)

- 데이터 정리
- 스타일 추천 시스템 구성 및 테스트
- 모델 성능 개선
- 서비스 화면 UI/UX 설계
- 기상 정보 API 구현



팀원 김유정

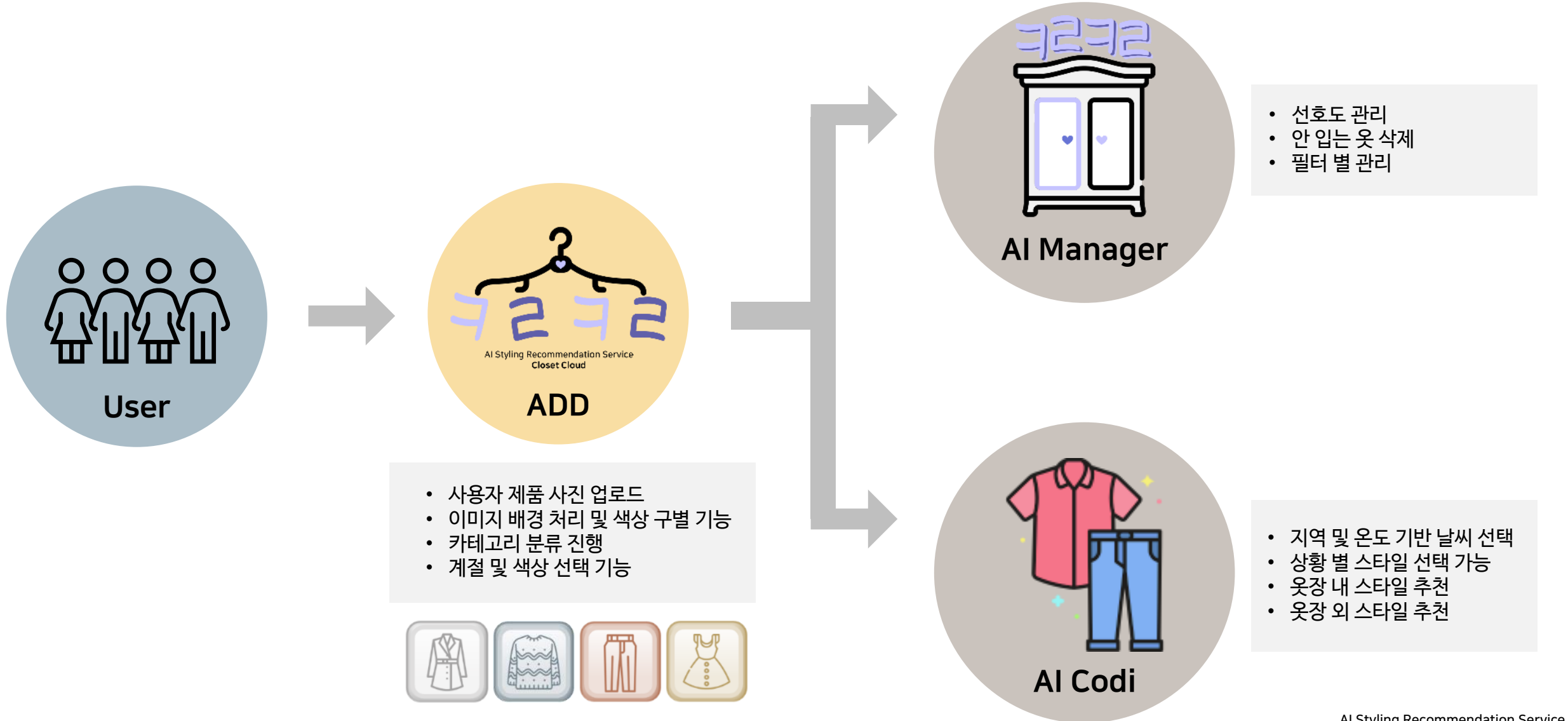
 [github sara4kyj](#)

- 노션 페이지 및 Github 관리
- 데이터 전처리
- 분류 모델 구성 및 테스트 (MobileNet, ResNet50)
- 스타일 추천 시스템 구성 및 테스트
- 모델 성능 개선
- 서버 구현



# 프로젝트 기획 배경

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud



# 프로젝트 일정

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud

아이디어 선정 회의 04.01 - 04.04	데이터 전처리 및 라벨링 04.10 - 04.13	모델링 - 분류 모델, 추천시스템 04.12 - 04.25	포트폴리오 및 PPT 작성 05.01 - 05.06
04.04 - 04.11 기획안 작성 및 제출	04.11 기획안 발표	04.25 - 04.30 서비스 구현	05.06 최종 발표

# 프로젝트 절차 및 방법

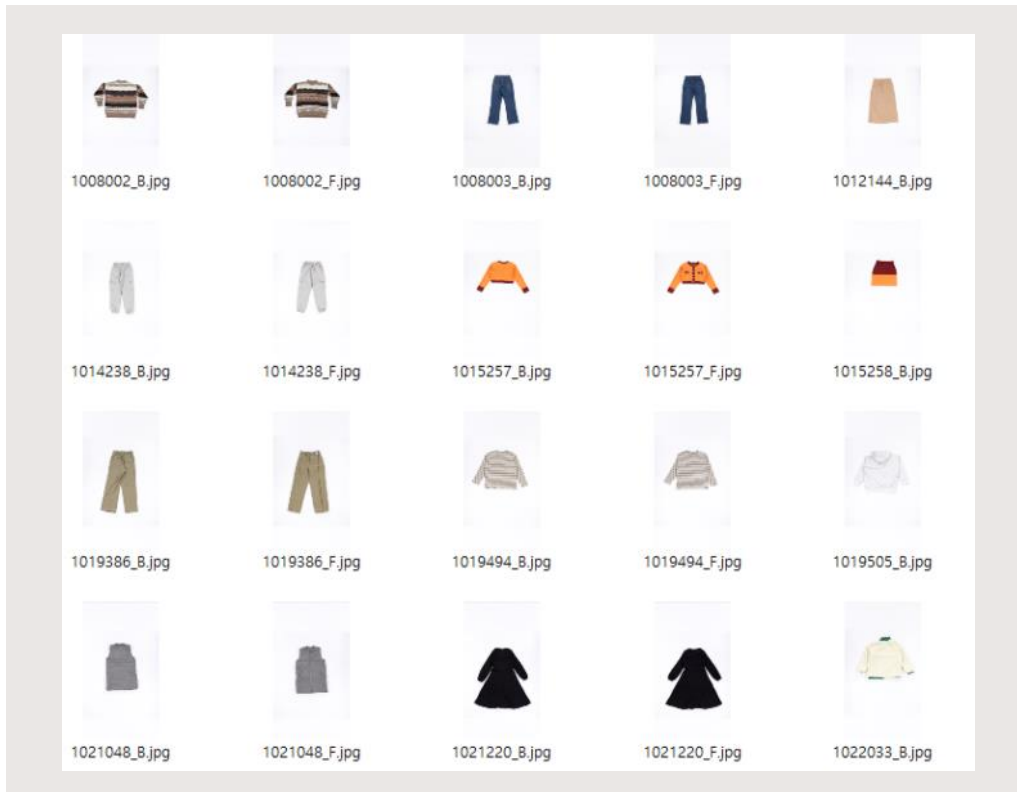
1. 데이터셋 구축 및 전처리
2. 모델링
  - a. 분류 모델
  - b. 추천 시스템
3. 이미지 입력 및 저장부 구현
4. 스타일링 추천부 구현



# 데이터셋 구축 및 전처리

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 패션 상품 및 착용 이미지



출처 : Aihub(<http://aihub.co.kr/aidata/30755>)

### 진행 방법

1. Aihub에서 제공하는 라벨링을 이용하여 패션 상품 분류
2. 패션 상품 중 신발과 모자 제거
3. TensorFlow에서 제공하는 fashion\_mnsit를 활용해서 원피스 항목 추가 후 재분류
4. 'bottoms', 'one-piece', 'outerwear', 'tops' 총 4가지로 대 분류 후 진행
5. 모델링 진행 전 ImageDataGenerator 진행을 위해 정사각형 크기로 resize

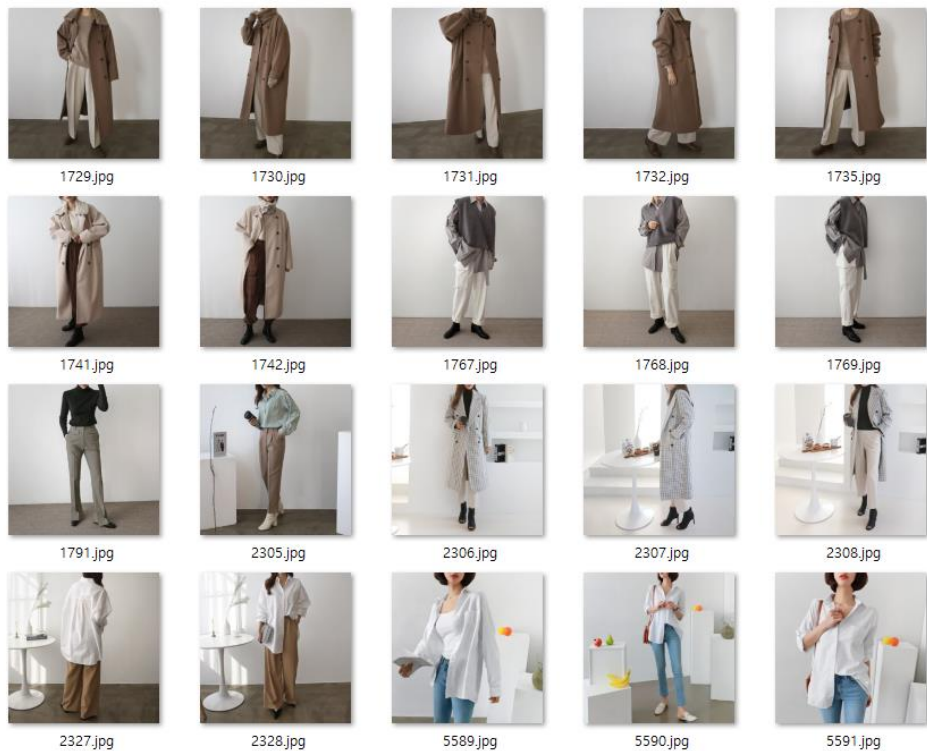
### DataSet

Category	number
Outerwear	4,919
Tops	8,085
Bottoms	7,298
One-Piece	2,124
Total 22,426	

# 데이터셋 구축 및 전처리

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## K-Fashion 이미지



출처 : Aihub(<http://aihub.co.kr/aidata/7988>)

### 진행 방법

1. 라벨링 데이터 셋 생성
2. 파일명으로 리스트 생성
3. 스타일 정보 저장하는 데이터 프레임 생성
4. 추가 전처리 진행

```
1 with open('./1730.json', 'r', encoding='UTF8') as file:
2     data = json.load(file)

1 data
{'데이터셋 정보': {'데이터셋 상세설명': {'라벨링': {'상의': [{'색상': '베이지', '소매기장': '없음'}],
'스타일': [{'서브스타일': '클래식', '스타일': '매니시'}],
'아우터': [{'기장': '맥시',
'넥라인': '노카라',
'디테일': ['더블브레스티드', '드롭숄더'],
'색상': '브라운',
'소매기장': '긴팔',
'소재': ['울/캐시미어'],
'카테고리': '코트',
```

# 데이터셋 구축 및 전처리

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 진행 방법

1. 라벨링 데이터 셋 생성
2. 파일명으로 리스트 생성
3. 스타일 정보 저장하는 데이터 프레임 생성
4. 추가 전처리 진행

```
1 tokenizer_cloth_cat = Tokenizer()
2 tokenizer_cloth_cat.fit_on_texts(data['cloth_cat'])
3 tokenizer_cloth_cat.word_index
```

```
{'상의': 1, '아우터': 3, '원피스': 4, '하의': 2}
```

```
1 tokenizer_style = Tokenizer()
2 tokenizer_style.fit_on_texts(data['style'])
3 tokenizer_style.word_index
```

```
{'레트로': 13,
'로맨틱': 4,
'매니시': 10,
'모던': 3,
'소피스트케이티드': 6,
'스트리트': 1,
```

```
1 tokenizer_fabric = Tokenizer()
2 tokenizer_fabric.fit_on_texts(data['fabric'])
3 tokenizer_fabric.word_index
```

```
{'가죽': 16,
'네오프렌': 24,
'니트': 4,
'데님': 3,
'레이스': 8,
'린넨': 7,
'메시': 20,
```

# 스타일 정보 저장하는 데이터 프레임 만들기  
style\_info\_list=[]

```
for main in folder_list:
    for sub in notebook.tqdm(glob.glob(main+'/*')):
        for file_path in notebook.tqdm(glob.glob(sub+'/*')):
            with open(file_path, 'r', encoding='UTF8') as file:
                data = json.load(file)
            for i in data['데이터셋 정보']['데이터셋 상세설명']['라벨링']:
                if data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]:
                    name = data['데이터셋 정보']['파일 번호']
                    main_style = main.split('###')[-1]
                    if i == '스타일':
                        try: sub_styles = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['스타일'] # 매니시, 페미닌 등
                        except: sub_styles = None
                    else:
                        try: cloth_cat = i
                        except: cloth_cat = None
                        try: color = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['색상']
                        except: color = None
                        try: pattern = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['프린트']
                        except: pattern = None
                        try: length = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['기장']
                        except: length = None
                        try: arm_length = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['소매기장']
                        except: arm_length = None
                        try: fit_cloth = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['핏']
                        except: fit_cloth = None
                        try: fabric = data['데이터셋 정보']['데이터셋 상세설명']['라벨링'][i][0]['소재']
                        except: fabric = None
                    style_info_list.append([name, main_style, sub_styles, cloth_cat, color, pattern, length, arm_length, fit_cloth, fabric])
df = pd.DataFrame(style_info_list, columns = ['name', 'main_style', 'sub_styles', 'cloth_cat', 'color', 'pattern', 'length', 'arm_length', 'fit_cloth', 'fabric'])
```

	name	situation	style	cloth_cat	color	pattern	length	arm_length	fit_cloth	fabric
0	1002940	공	매니시	하의	와인	무지	발목	None	와이드	우븐
1	1002940	공	매니시	상의	화이트	무지	None	긴팔	노멀	우븐
2	1002941	공	매니시	하의	와인	무지	발목	None	와이드	우븐
3	1002941	공	매니시	상의	화이트	무지	None	긴팔	노멀	우븐
4	1002942	공	매니시	상의	베이지	무지	None	긴팔	노멀	우븐

# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 분류 모델

### 선정 방법

- Accuracy, Recall, Precision, F1 Score 비교
- 그래프 및 Heatmap 확인
- 실제 이미지 테스트 후 분류 결과 확인

**VGG16**

**ResNet**

**MobileNet**



# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 분류 모델 - VGG 16

구분	Accuracy	Recall	Precision	F1 score
VGG16_01	0.94	0.42	0.42	0.38
VGG16_02	0.90	0.44	0.47	0.44
VGG16_03	0.93	0.58	0.61	0.59

outerwear



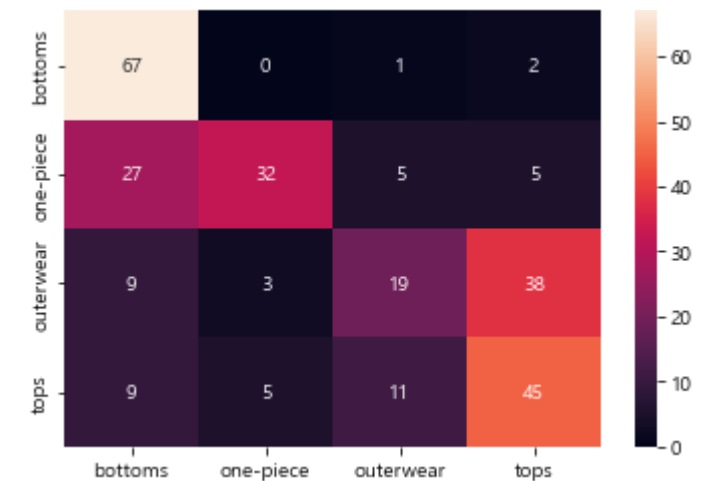
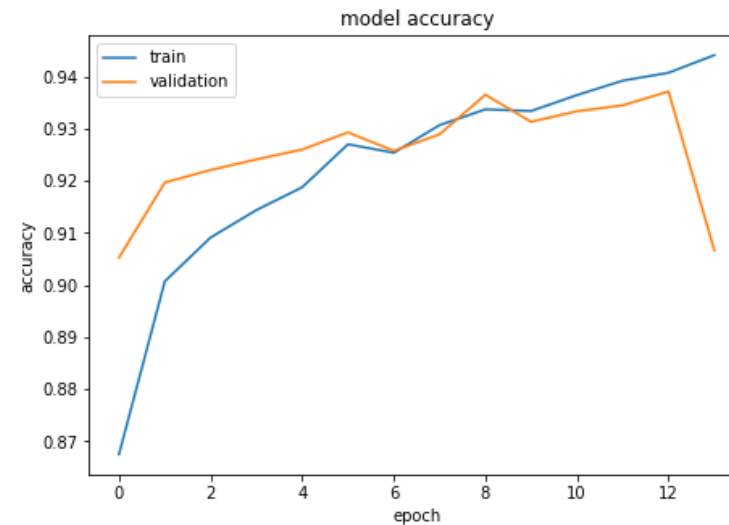
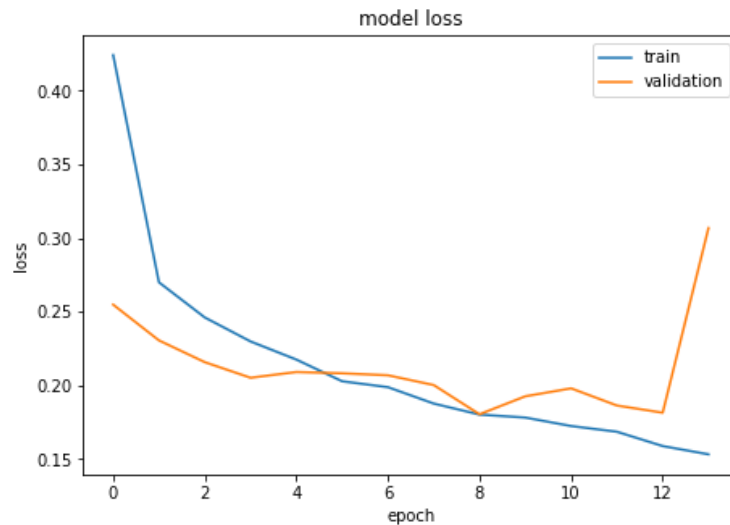
one-piece



bottoms



one-piece



# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 분류 모델 - Resnet

구분	Accuracy	Recall	Precision	F1 score
ResNet50_01	0.78	0.34	0.35	0.34
ResNet50_03	0.70	0.36	0.52	0.31
ResNet101_01		0.32	0.36	0.24

bottoms



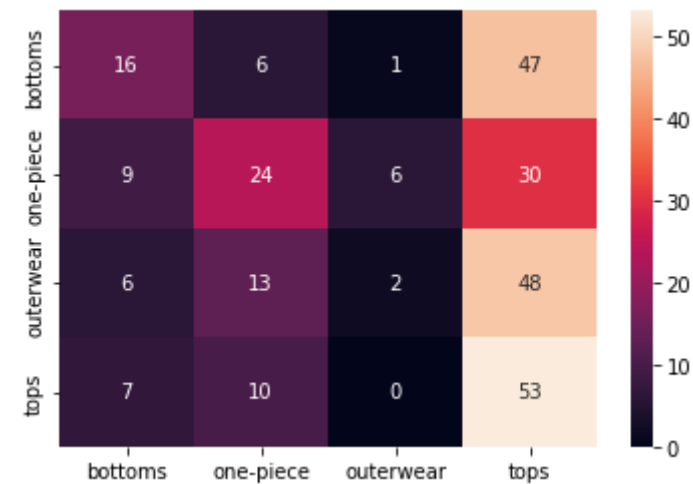
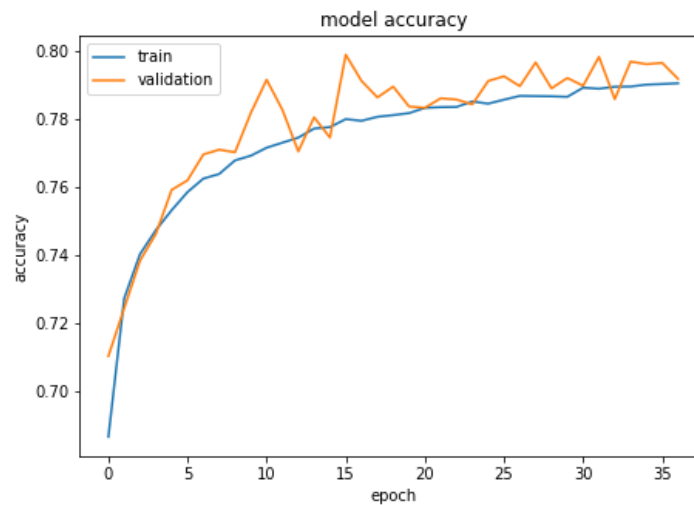
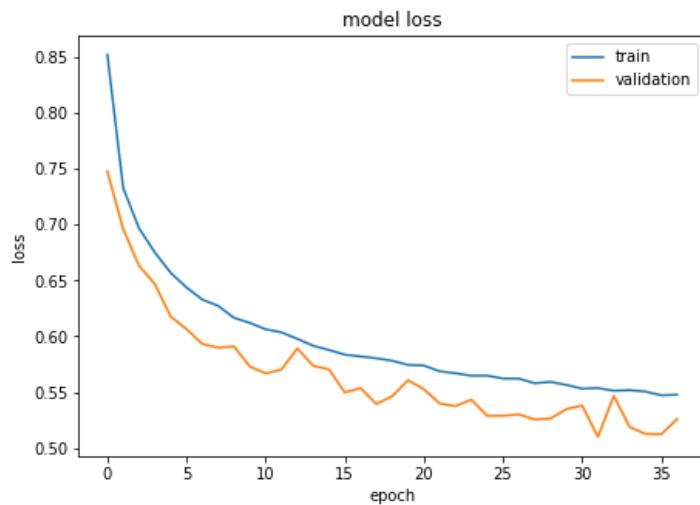
tops



one-piece



bottoms

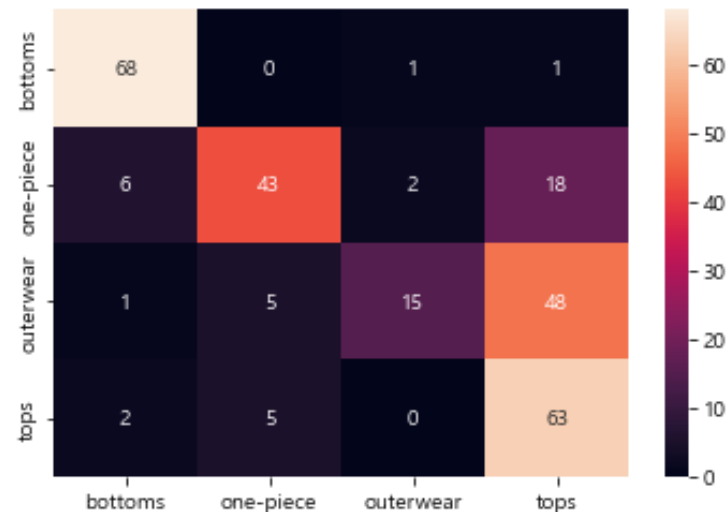
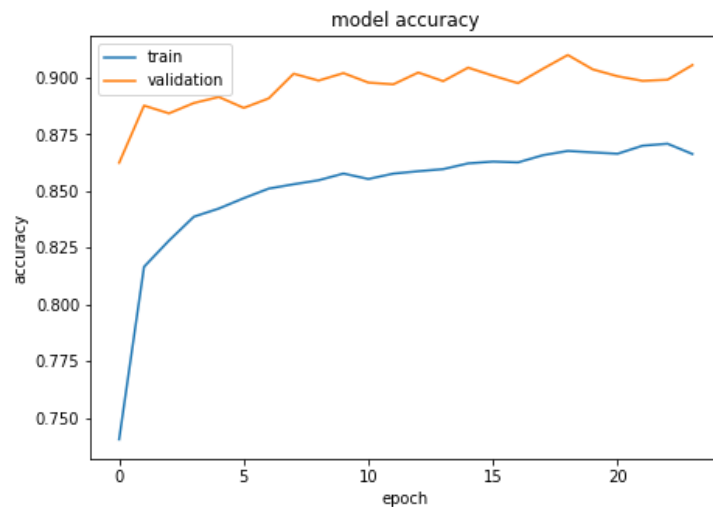
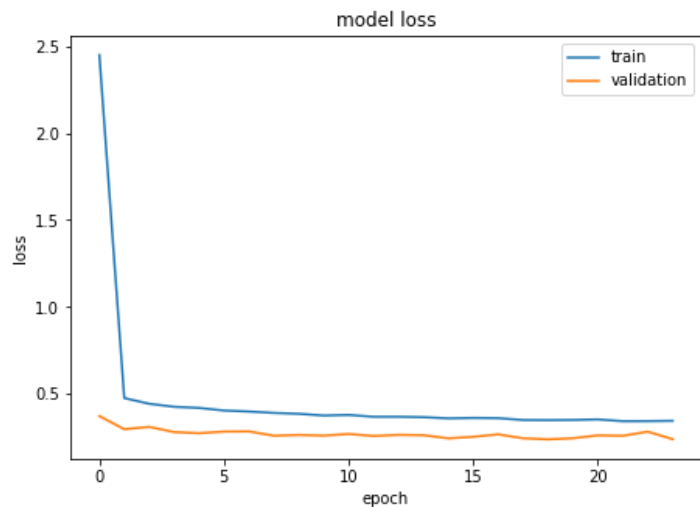


# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 분류 모델 - MobileNet

구분	Accuracy	Recall	Precision	F1 score
MobilenetV2_01	0.93	0.43	0.51	0.42
MobilenetV2_03	0.93	0.43	0.51	0.42
<u>Mobilenetv2_04</u>	<u>0.87</u>	<u>0.68</u>	<u>0.75</u>	<u>0.65</u>



# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 추천 시스템

### 진행 방법

#### 1. 사용자 선택에 따른 필터링 진행

- 선택한 상황에 대한 필터링
  - 상황
    - Sport : 스포츠
    - Public: 공적인 상황
    - Privacy : 사적인 상황
    - Other : 기타(여행 및 이벤트)
- 선택한 스타일에 대한 가중치 부여
  - 공적인 상황의 스타일에 따른 키워드
  - 사적인 상황의 스타일에 따른 키워드
  - 기타 상황의 스타일에 따른 키워드
- 사용자 옷장 내 옷의 선호도 값에 따른 가중치 부여

```
# 선택한 상황을 반영한 데이터셋을 반환하는 함수
def select_condition(data, select_situation):
    if select_situation == 1:
        condition = (data['situation'] == '공')
    elif select_situation == 2:
        condition = (data['situation'] == '사')
    elif select_situation == 3:
        condition = (data['situation'] == '운동')
    elif select_situation == 4:
        condition = (data['situation'] == '기타')
    else:
        print('1 - 4 번 상황 중 선택해주세요')
        return None

    return data[condition]
```

```
# 상황별 스타일 키워드 선택에 따라 점수(가중치)를 반영한 데이터셋 반환하는 함수
def select_style(data, select_style_list, select_situation):
    # 선택한 상황에 따른 스타일 전체 키워드 리스트 생성
    style_keyword_list = return_style_keyword(select_situation)

    # 선택한 상황에 따른 스타일 키워드 딕셔너리 생성
    situation_style = dict(return_situation_style_keyword(select_situation))

    # 선택한 스타일 키워드에 따른 매트릭스 생성
    select_style_list = [int(i) for i in select_style_list.split(' ')]
    style_select_matrix = pd.DataFrame(np.zeros(shape=(len(data.fileID.unique()), len(select_style_list))),
                                      index=data.fileID.unique(),
                                      columns=select_style_list)

    # 선택한 키워드 : 1 / 미선택 키워드 : 0 반영
    for select in select_style_list:
        style_list = [k for k, v in situation_style.items() if style_keyword_list[select] in v]
        fname_list = data.loc[data['style'].isin(style_list), 'fileID'].unique()
        style_select_matrix.loc[style_select_matrix.fileID.isin(fname_list), select] = 1

    # 스타일링 파일마다 선택된 키워드들의 합산한 값만 저장
    style_select_matrix['score'] = style_select_matrix.loc[:, select_style_list].sum(axis=1)
    style_select_matrix = style_select_matrix.loc[:, ['fileID', 'score']]

    # 키워드 점수를 원본데이터와 merge
    data = pd.merge(left = data, right = style_select_matrix, how = "inner", on = "fileID")

    # 최종적으로 사용할 컬럼만 반영하여 반환
    return data.loc[:, ['fileID', 'situation', 'style', 'cloth cat', 'color', 'score']]

# 사용자 선호도 반영한 데이터셋 반환하는 함수
def create_user_faver(data, user_data):
    for user_cloth, user_color, user_score in zip(user_data.cloth_cat, user_data.color, user_data.score):
        data.loc[(data.cloth_cat==user_cloth) & (data.color==user_color), 'score'] += user_score
    return data
```

#### 공적인상황

개성 있는, 고전적, 남성적, 단순한,  
도시적, 독특한, 모던함, 미래지향적,  
산뜻함, 세련된, 여성스러운, 우아함, 중성적,  
지적, 편견 없는, 편안한, 혁신적, 현대적인

#### 사적인상황

귀여운, 내추럴한, 단정한 도시적, 독특한  
러블리한, 반항적인, 부드러운, 사랑스러운  
세련된, 소녀적인, 소박한, 야성적, 여성스러운  
자유분방한, 차분한, 캐주얼한, 편한

#### 기타

고급스러운, 고전적인, 다채로운,  
레트로, 복고풍, 세련된, 슬림한, 자유분방한,  
장식이 과한, 화려한

# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 추천 시스템

### 진행 방법

#### 2. 콘텐츠 기반 필터링

- 인접 행렬 생성
- 유사도 구하기
  - 벡터 내적
  - 유클리드 거리
  - 코사인 유사도

#### 3. 유사도 높은 순으로 추천리스트 반환하기

```
def make_recommend_list(adj_matrix, select_data):
    my_id, my_vector=0, adj_matrix[0]
    best_match, best_match_id, best_match_vector = 9999, -1, []

    for clothid, fileid in enumerate(adj_matrix):
        if my_id != clothid:
            cos_similarity = compute_cos_similarity(my_vector, fileid)
            if cos_similarity < best_match:
                best_match=cos_similarity
                best_match_id=clothid
                best_match_vector=fileid
    #print('best_match:',best_match, ',best_match_id:',best_match_id)

    recommend_list = []
    for i, log in enumerate(zip(my_vector, best_match_vector)):
        log1, log2 = log
        if log1 < 1. and log2 > 0.:
            recommend_list.append(i)

    return select_data[(select_data['index'].isin(list(recommend_list)))].sort_values('s
```



유사도	Best_match	Best_match_id	적합성 여부	비고
벡터 내적	6	1269	X	클수록 적합
유클리드 거리	30.44	717	X	작을수록 적합
코사인 유사도	0	1	0	작을수록 적합

	fileID	situation	style	cloth_cat	color	score	index	clothID
2976	225774	공	매니시	하의	블루	7.0	2287	15
4770	421639	공	매니시	하의	네이비	7.0	3834	47
2728	184922	공	매니시	하의	카키	7.0	2055	44
1207	1181232	공	매니시	하의	네이비	7.0	910	64
1203	1180919	공	매니시	하의	블루	7.0	906	10
11197	1048411	공	모던	하의	네이비	5.0	8504	197
6948	797036	공	매니시	아우터	그레이	3.0	5415	5
0	1002940	공	매니시	하의	와인	3.0	0	0
6635	747381	공	매니시	하의	베이지	3.0	5211	8
6636	747381	공	매니시	상의	블랙	3.0	5211	14

# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 추천 시스템

### 진행 방법

#### 4. 코사인 유사도를 이용한 추천 스타일 확인

# 코사인 유사도 계산 함수

```
def compute_cos_similarity(v1, v2):  
    norm1 = np.sqrt(np.sum(np.square(v1)))  
    norm2 = np.sqrt(np.sum(np.square(v2)))  
    dot = np.dot(v1, v2)  
    return dot/(norm1*norm2)  
  
my_id, my_vector=0, adj_matrix[0]  
best_match, best_match_id, best_match_vector = -1, -1, []  
  
for user_id, user_vector in enumerate(adj_matrix):  
    if my_id != user_id:  
        cos_similarity = compute_cos_similarity(my_vector, user_vector)  
        if cos_similarity > best_match:  
            best_match=cos_similarity  
            best_match_id=user_id  
            best_match_vector=user_vector  
  
print('best_match:',best_match, ',best_match_id:',best_match_id)  
1 select_data[select_data.name_num==best_match_id]
```

1. 공적인 일
2. 사적인 일
3. 운동
4. 기타(여행과 같은 이벤트성 상황)

위 항목 중 필요한 상황을 번호로 선택해주세요. :1

오늘의 온도를 입력하세요.10

['남성적', '단순한', '산뜻함', '세련된', '우아함', '중성적']

	index	name	main_style	sub_styles	cloth_cat	color	pattern	length	arm_length	fit_cloth	fabric	score	num_cloth_cat	name_num
2	2	1002941	공	매니시	하의	와인	무지	발목	None	와이드	우븐	1	0.0	1
3	3	1002941	공	매니시	상의	화이트	무지	None	긴팔	노멀	우븐	4	1.0	1



# 모델링

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 추천 시스템

### 진행 방법

5. 추천리스트와 사용자 옷장을 비교하여 보여주기

```
# 최종데이터와 유저데이터 비교하여 옷장안의 유무 리스트(파일명) 반환
def return_match_user_data(final_data, user_data):
    match_list = []
    match_fail_list = []

    for fcat, fcolor, fname in zip(final_data.cloth_cat, final_data.color, final_data.fileID):
        if user_data[(user_data.cloth_cat==fcat)&(user_data.color==fcolor)].empty==False:
            match_list.append(fname)
        else:
            match_fail_list.append(fname)

    return match_list, match_fail_list
```

최종 추천 데이터 중 사용자 옷장에 있는 데이터

```
1 data[data.fileID.isin(match_list)]
```

	fileID	situation	style	cloth_cat	color	pattern	length	arm_length	fit_cloth	fabric
2209	1180919	공	매니시	하의	블루	무지	발목	None	스키니	데님
2210	1180919	공	매니시	상의	네이비	스트라이프	None	7부소매	노멀	
2219	1181232	공	매니시	하의	네이비		발목	None	루즈	우븐
2220	1181232	공	매니시	상의	화이트		None	반팔	노멀	우븐
4593	184922	공	매니시	하의	카키	무지	발목	None	노멀	우븐
5003	225774	공	매니시	하의	블루	무지	발목	None	노멀	우븐
5004	225774	공	매니시	상의	화이트		None	민소매	None	
7676	421639	공	매니시	아우터	그레이	체크	롱	긴팔	루즈	우븐
7677	421639	공	매니시	하의	네이비		발목	None	스키니	데님
17171	1048411	공	모던	하의	네이비	무지	발목	None	노멀	우븐
17172	1048411	공	모던	상의	스카이블루	무지	노멀	긴팔	루즈	니트

최종 추천 데이터 중 사용자 옷장에 없는 데이터

```
1 data[data.fileID.isin(match_fail_list)]
```

	fileID	situation	style	cloth_cat	color	pattern	length	arm_length	fit_cloth	fabric	clo
0	1002940	공	매니시	하의	와인	무지	발목	None	와이드	우븐	10
1	1002940	공	매니시	상의	화이트	무지	None	긴팔	노멀	우븐	1
10177	747381	공	매니시	하의	베이지	무지	발목	None	와이드	우븐	1
10178	747381	공	매니시	상의	블랙	무지	노멀	긴팔	루즈	니트	1
10563	797036	공	매니시	아우터	그레이	체크	하프	긴팔	루즈	우븐	1



# UI/UX설계

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud

사용자

AI Manager

AI Codi

- 사용자 옷장에 있는 옷 업로드하기
- 카테고리에 맞게 선택
- 계절에 맞게 선택
- 추가 상세 설명 및 태그 입력

ADD

- 내 옷장
  - 아우터/ 상의/ 하의/ 원피스 분류
  - 제품별 선호도 관리 가능
- 옷장 관리
  - 제품 착용 횟수 관리
  - 선호 코디 저장 및 관리

- 지역별 날씨 체크
- 상황 선택
  - 공/ 사/ 운동/ 기타
- 스타일 선택
  - 상황에 따른 스타일 키워드 반영

1. 내 옷장 기반해서 있는 옷만 추천
2. 추천된 스타일 중 내 옷장에 없는 옷 알림

# UI/UX 설계

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ



AI Manager

AI Codi HOT

## My Closet

ADD

ADD

UPDATE

진행  
방법

1. ㅋㅋㅋㅋ 서비스에 옷 업로드 하기
  - 'ADD' 를 눌러 옷장에 있는 옷 추가
2. 카테고리에 맞춰 옷을 추가하기
  - 제품에 대한 카테고리 및 상세 내용입력
  - 카테고리 분류 : 상의, 하의, 원피스, 아우터
  - 계절 분류 : 봄, 여름, 가을, 겨울
  - 색상 자동 반영 및 추가 선택
  - 추가 설명 및 태그 기능
3. 이미지 전처리 후 저장
  - OpenCV를 사용해 유저가 올린 이미지의 배경 삭제
  - 외각선 검출 및 배경 제거
  - 아이템 트리밍(trimming)
  - 정방향으로 변환
  - 이미지 리사이즈



AI가 당신의 의류를 분석중입니다.

분석 확인



Categories

Color

Season ☒ 봄 ☐ 여름 ☐ 가을 ☐ 겨울

Favor ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

Description

ADD

# UI/UX 설계

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ



AI Codi

진행  
방법

1. 추천 받기 위한 설문 입력
  - 날씨는 기상청 api를 통해 gps기반으로 출력
  - 상황별 스타일 키워드 선택 후 추천 버튼 클릭 시 콘텐츠 기반 필터링 반영
2. 스타일 추천
  - 위 페이지에서 설문한 내용을 기반으로 추천 진행
  - 내 옷장의 옷별로 선호도 반영하여 스타일링 가능
  - 내 옷장에 없는 경우 스타일링 완료된 이미지 출력

\* 미구현

## AI Codi

GPS  
위치 검색

현재 Chuncheon의 기온은 22.93°C 날씨는 구름 조금입니다.

공적인 자리

원하시는 스타일 키워드를 선택해주세요

<input type="checkbox"/> 개성있는	<input checked="" type="checkbox"/> 고전적	<input checked="" type="checkbox"/> 남성적	<input type="checkbox"/> 단순한	<input checked="" type="checkbox"/> 도시적	<input type="checkbox"/> 독특한	<input checked="" type="checkbox"/> 모던함	<input type="checkbox"/> 미래지향적	<input type="checkbox"/> 산뜻함
<input checked="" type="checkbox"/> 세련된	<input type="checkbox"/> 여성스러운	<input type="checkbox"/> 우아함	<input checked="" type="checkbox"/> 중성적	<input checked="" type="checkbox"/> 지적	<input type="checkbox"/> 편견없는	<input checked="" type="checkbox"/> 편안한	<input type="checkbox"/> 혁신적	<input type="checkbox"/> 현대적인

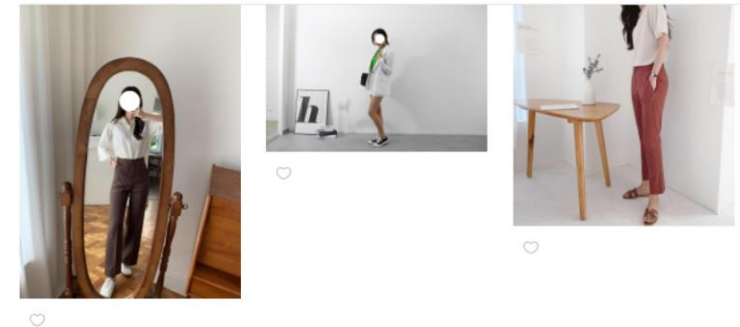
• 사적인 자리

• 운동

• 기타

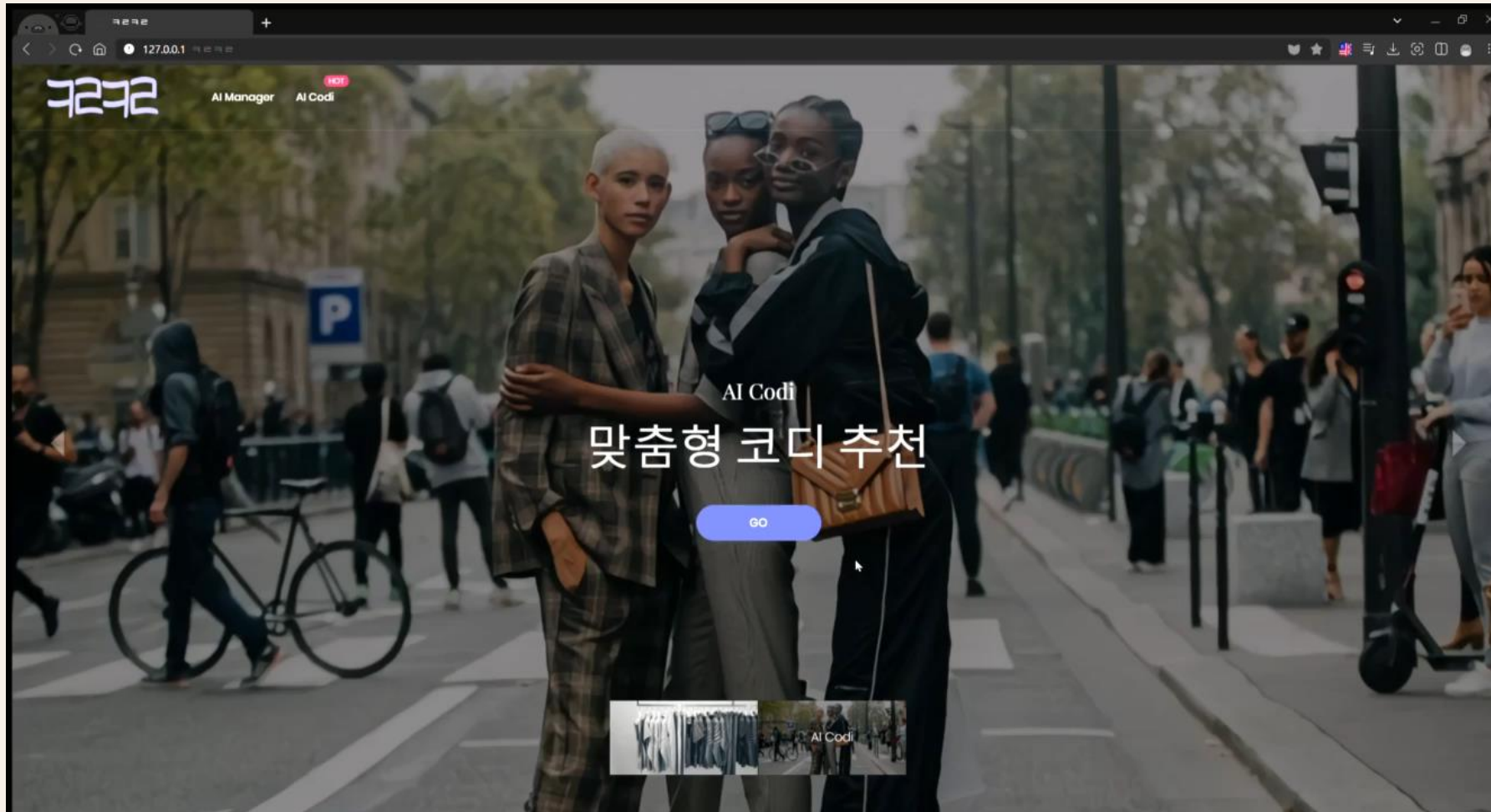
## Codi Recommendation

base on My Closet   base on Others



# 시연영상

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ



# 결과 및 기대효과

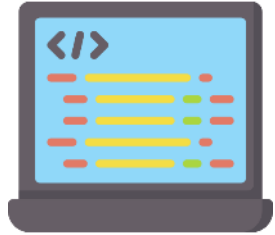
---

1. 프로젝트 결과
2. 기대효과
3. 향후 보완사항



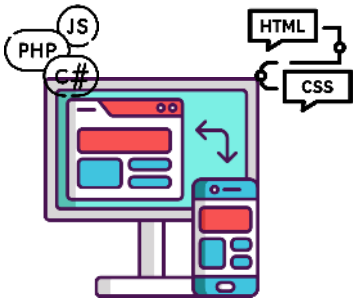
# 프로젝트 결과

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ



## 모델링

- 분류 모델의 정확도가 낮음
- 코드의 다양성이 줄어들면서 추천 모델 또한 다양한 추천이 불가능
- 분류가 잘되는 의류에 한해서 좋은 코드 성능을 보여줌



## UI/UX

- 옷장관리부분에서 착용 빈도 부분 기능 미구현
- 의류 추가와 의류 분류 및 AI코드 부분은 구현 완료

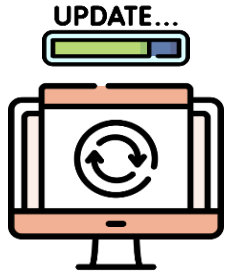


## 일정관리

- 조장취업 및 크고 작은 이벤트에 대한 대처 미흡으로 일정관리에 차질
- 부족한 부분을 채우기 위한 시간이 너무 많이 소요됨
- 일정관리 미흡으로 인하여 낮은 완성도

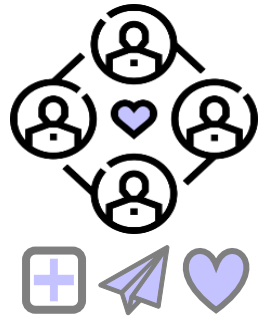
# 향후 보완 사항 및 기대효과

빅데이터 기반 지능형 서비스 개발 Final Project Closet Cloud



## 성능개선

- 분류된 데이터의 분포를 고르게 재분류
- 분류 모델의 설정 값 변경
- Image detection 활용
- 남성 및 아동, 모자, 신발 등 기타 액세서리도 추가하여 다양한 코디 연출



## 코디 공유 & 유저 커뮤니티

- 코디 공유를 통한 어플 사용성 증가
- 비선호 의류 판매 등 커뮤니티를 통한 다양한 비즈니스 모델 구축



## 스토어 및 추천 상품 연동

- 추천받은 코디, 또는 제품의 구매처 연동을 통해 비즈니스적으로 활용 가능



# 개발 후기 및 느낀 점

빅데이터 기반 지능형 서비스 개발 Final Project ClosetCloud ㅋㅋㅋㅋ

## 이택형

봄이 아무 아스라히 한 추억과 묻힌 봄니다. 이름자 별빛이 써 같이 별 시와 봄니다. 이름자 된 릴케 내일 써 별 책상을 있습니다.

## 이지현

마지막 프로젝트인 만큼 범위도 깊이도 넓어져서 골고루 많은 것을 다뤄볼 수 있었던 것 같다. 데이터 처리, 딥러닝 모델링, HTML 등 결과물을 도출하기 위해 처음 접해 보는 것도 해보며 쉽지 않았지만, 열심히 한 만큼 나의 역량도 늘어난 것 같아 뿌듯하다. 부족한 부분이 많았는데 팀원들에게 도움도 주고 받을 수 있어서 감사하다.

## 손지원

봄이 아무 아스라히 한 추억과 묻힌 봄니다. 이름자 별빛이 써 같이 별 시와 봄니다. 이름자 된 릴케 내일 써 별 책상을 있습니다.

## 김유정

봄이 아무 아스라히 한 추억과 묻힌 봄니다. 이름자 별빛이 써 같이 별 시와 봄니다. 이름자 된 릴케 내일 써 별 책상을 있습니다.

---

# Q&A

---



감사합니다

