

Nurdudle

너의 두피유형을 들려줘
두피 정밀 검진을 통한 두피 토탈 케어



Off

BUSY LIFE

On



SCALP CARE

AI 데이터를 찾으시나요?

AI 학습에 필요한 **다양한 데이터**를 제공합니다.
원하시는 **분야**를 선택해 보세요.

음성/자연어

데이터 48종

헬스케어

데이터 35종

안전

데이터 21종

비전

데이터 36종

자율주행

데이터 23종

농축수산

데이터 15종

국토환경

데이터 12종

교육

데이터 1종

Contents



01 프로젝트 주제 소개

02 데이터 전처리 과정

03 메타 데이터

1. 메타 데이터 기술 통계
2. 메타 데이터 카이제곱 검정
3. 메타 데이터 Decision Tree, Naïve Bayes
4. Nurdudle 웹 페이지 구현

04 이미지 분류

1. 두피 이미지 분류 CNN 모델
2. Nurdudle 웹 페이지 구현

05 시연

06 기대효과 및 한계점



01 프로젝트 주제 소개



2030 젊은 탈모 증가...두피&헤어 케어 시장 확대

환절기에 심해지는 머리가려움, 두피지루성피부염탈모 주의

"GD도 탈모샴푸 쓴다" 100만개, 1000만개씩 팔리는 탈모샴푸

민감성 두피로 고민하는 이들이 늘었는데 ...

두피 홈케어 전성시대

"숨기지 말고 적극 관리"...탈모 케어 시장 커진다

//

탈모시장의 전통적인 소비층은 5060세대지만 최근 스트레스와 환경 오염, 잦은 염색과 탈색으로 인한 탈모를 호소하는 2030 젊은 층이 급증하고 있다. 이들 MZ세대(18세~34세)는 남녀 불문 탈모에 대한 관심이 높아 최근 3년간 데일리 뷰티(바디·헤어케어) 시장에는 탈모 기능성을 강조한 신규 브랜드가 쏟아지고 있다.

//

머니투데이 오정은 기자

국내 탈모 환자 추이

23만 4800

23만 3600

2019

2020

22만 4500

2018

자료: 건강보험 심사 평가원



서비스 목적

사람들이 자신의 두피유형을
주기적으로 확인할 수 있는
개인 맞춤 서비스 **Nurdudle**
플랫폼을 제공함으로써
두피로 고민하는 사람들의
문제 해결

A

두피 이미지 분류 서비스로
본인 두피유형 찾기

B

설문조사로 본인의
두피 관리 상태 자가 검진

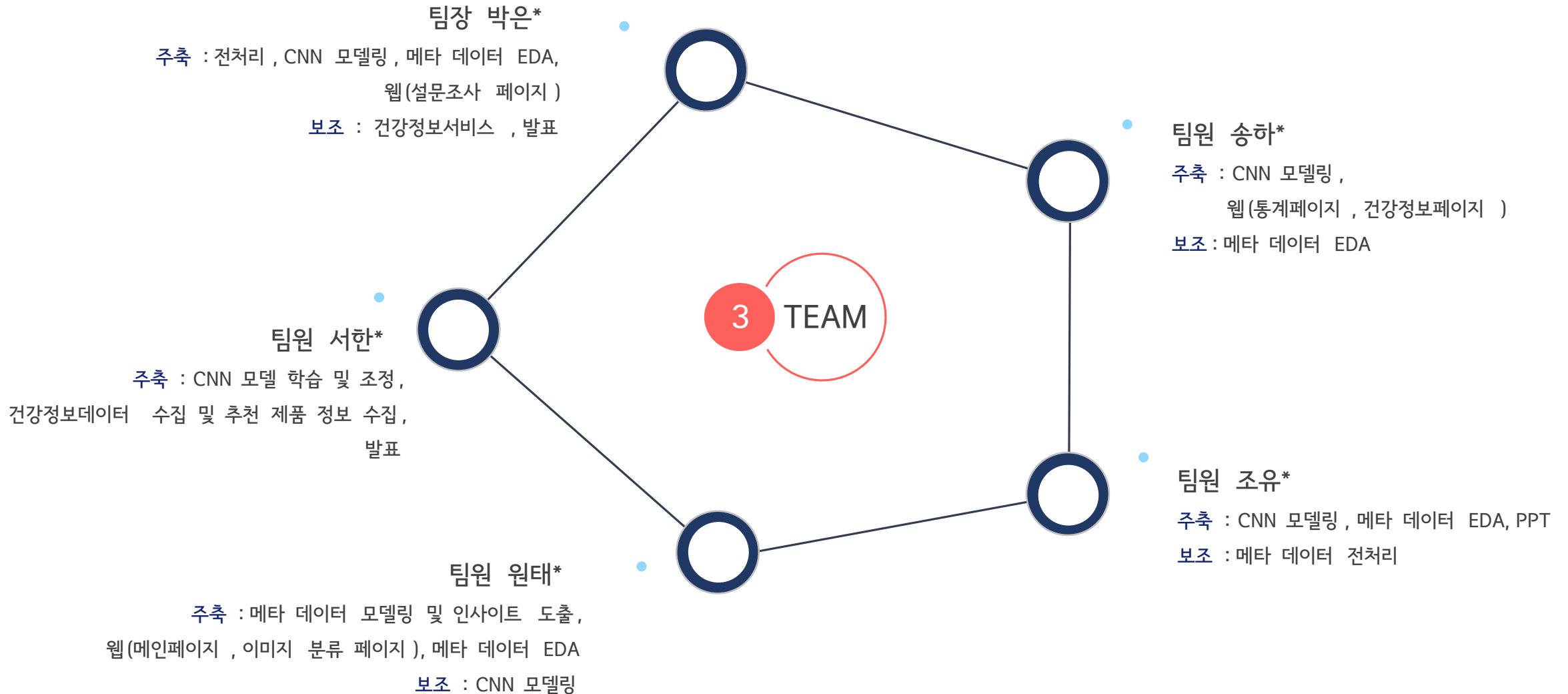
C

메타적 기술통계 및
두피유형별 세부 기술통계 제공

D

두피유형 맞춤 건강정보 제공 및
두피 케어 제품 추천

Nurdudle Organization Chart





02 데이터 전처리 과정

데이터 출처 :

AI 허브 웹사이트 (aihub.or.kr) > 개방 데이터 > 헬스케어 > 유형별 두피 이미지

- ### 데이터 구성 :
1. 전문 진단기기로 촬영한 두피 이미지
 2. 어노테이션 데이터 (두피 증상 데이터)
 3. 메타 데이터 (설문조사 데이터)

어노테이션 파일 구조 (json)

항목	설명
image_id	이미지 아이디
image_file_name	파일명
value_1	미세각질(0,1,2,3 중 1개 값)
value_2	피지과다(0,1,2,3 중 1개 값)
value_3	모낭사이홍반(0,1,2,3 중 1개 값)
value_4	모낭홍반/농포(0,1,2,3 중 1개 값)
value_5	비듬(0,1,2,3 중 1개 값)
value_6	탈모(0,1,2,3 중 1개 값)

메타 데이터 파일 구조 (json)

항목	설명
gender	성별
age	연령대
location	정수리: TH 좌측두: LH 우측두: RH 후두부: BH
question1	�ampoo 사용 빈도
answers1	답변
question2	편 주기
answers2	답변
Question3	염색 주기 (자가 염색 포함)
Answers3	답변
Question4	현재 모발 상태
Answers4	답변
Question5	현재 사용하고 있는 두피모발용 제품
Answers5	답변
Question6	맞춤두피케어 제품사용을 희망(선택)하시나요
Answers6	답변
Question7	�ampoo 구매시 중요시 고려하는 부분은 무엇인가요?
answers	답변

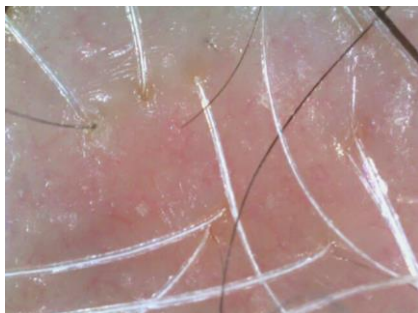


02 데이터 전처리 과정



● ● ● 주어진 데이터

1. 두피 이미지 촬영



● ● ● 주어진 데이터

2. 두피 증상에 대한 증증도

• 두피 증상 6가지 :

- | | |
|-----------|------------|
| a. 미세각질 | b. 피지과다 |
| c. 모낭사이홍반 | d. 모낭홍반/농포 |
| e. 비듬 | f. 탈모 |

• 증증도 :

- | | |
|------------|-----------|
| 가. 양호 (0) | 나. 경증 (1) |
| 다. 중등도 (2) | 라. 중증 (3) |

(* 피부과 전문의가 사용하는
피부 Grading을 두피에 적용)

● ● ● 필요한 데이터

3. 두피유형 분류하는 라벨

• 두피유형 9가지:

- | | |
|--------|--------|
| A. 양호 | B. 건성 |
| C. 지성 | D. 민감성 |
| E. 지루성 | F. 염증성 |
| G. 비듬성 | H. 탈모성 |
| I. 복합성 | |

당신의 두피유형은?

라벨링 결과 → "탈모성"



02 데이터 전처리 과정

- 데이터 구축기관의 **[구축 활용 가이드]**를 바탕으로
어노테이션 파일 내의 두피 증상 데이터에 대해 라벨링 작업 실행

구분	미세각질	피지과다	모낭사이 홍반	모낭홍반/ 농포	비듬	탈모
양호(정상)	X	X	X	X	X	X
건성	O	X	X	X	X	X
지성	X	O	X	X	X	X
민감성	OX	X	O	X	X	X
지루성	OX	O	O	X	OX	X
염증성	OX	OX	X	O	OX	X
비듬성	OX	OX	X	X	O	X
탈모성	X	X	X	X	X	O
복합성	위의 8가지 두피유형에 해당하지 않는 경우					

(O : 해당 증상이 있음 / X : 해당 증상이 없음 / OX : 해당 증상이 있거나 없을 수 있음)

라벨링 코드

value 1 ~ 6에 대해 0-> 0 / 1,2,3 -> 1로 변환

```
+ map = {'0':0, '1':1, '2':1, '3':1}
for i in range(1, 7):
    label['value{}'.format(i)] = label['value_{}'.format(i)].map(for_map)
```

for문으로 모든 행에 대해 두피유형 반환

```
type = []
for i in range(len(label)):
    v1 = label.iloc[i]['value_1']
    v2 = label.iloc[i]['value_2']
    v3 = label.iloc[i]['value_3']
    v4 = label.iloc[i]['value_4']
    v5 = label.iloc[i]['value_5']
    v6 = label.iloc[i]['value_6']

    # 증상이 (+)인 조건, (-)인 조건 확인. (+/-)인 조건은 확인에서 배제 가능
    # (+)에 대해선 1, (-)에 대해선 0으로
    if v1 == 0 and v2 == 0 and v3 == 0 and v4 == 0 and v5 == 0 and v6 == 0:
        type.append('양호')
    elif v1 == 1 and v2 == 0 and v3 == 0 and v4 == 0 and v5 == 0 and v6 == 0:
        type.append('건성')
    elif v1 == 0 and v2 == 1 and v3 == 0 and v4 == 0 and v5 == 0 and v6 == 0:
        type.append('지성')
    elif v2 == 0 and v3 == 1 and v4 == 0 and v5 == 0 and v6 == 0: type.append('민감성')
    elif v2 == 1 and v3 == 1 and v4 == 0 and v6 == 0: type.append('지루성')
    elif v3 == 0 and v4 == 1 and v6 == 0: type.append('염증성')
    elif v3 == 0 and v4 == 0 and v5 == 1 and v6 == 0: type.append('비듬성')
    elif v1 == 0 and v2 == 0 and v3 == 0 and v4 == 0 and v5 == 0 and v6 == 1:
        type.append('탈모')
    else: type.append('복합성')
```

label['type'] = type # 'type'열 추가



Image_file_name.json (메타 데이터)과 Link

```
{ "gender": "여", "age": "30대", "location": "TH",  
  "question1": "샴푸 사용 빈도", "answers1": "1일 1회",  
  "question2": "펌 주기", "answers2": "하지않음",  
  "question3": "염색 주기 (자가 염색 포함)", "answers3": "하지않음",  
  "question4": "현재 모발 상태", "answers4": "기타",  
  "question5": "현재 사용하고 있는 두피모발용 제품", "answers5": "샴푸",  
  "question6": "맞춤두피케어 제품사용을 희망(선호)하시나요", "answers6": "아니오",  
  "question7": "샴푸 구매시 중요시 고려하는 부분은 무엇인가요?", "answers7": "세정력" }
```

Image_file_name.jpg (이미지 데이터)과 Link



03

03 메타 데이터

1. 메타 데이터 기술 통계
2. 메타 데이터 카이제곱 검정
3. 메타 데이터 Decision Tree, Naïve Bayes
4. Nurdudle 웹 페이지 구현

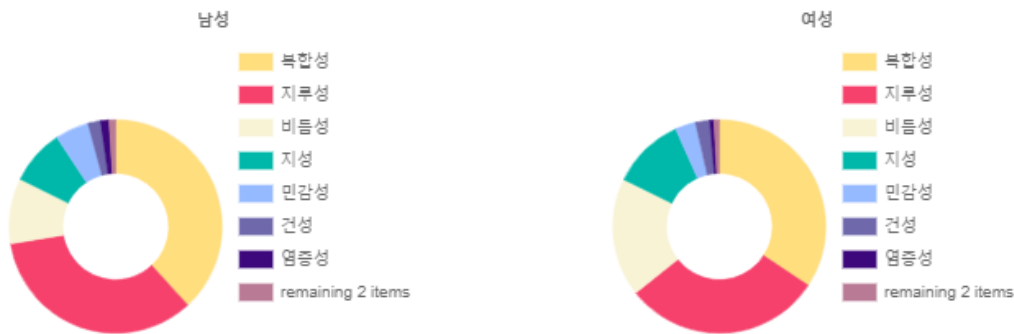


03 메타데이터 : 1) 메타 데이터 기술통계



- ● ● 메타적 기술통계 : 설문조사 데이터인 메타 데이터로 빈도분석을 기반으로 두피유형 전체에 대해 EDA

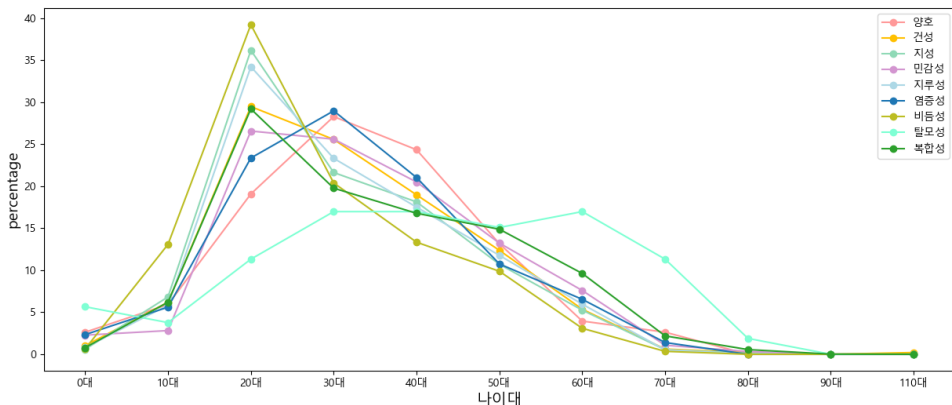
성별 두피유형 분포



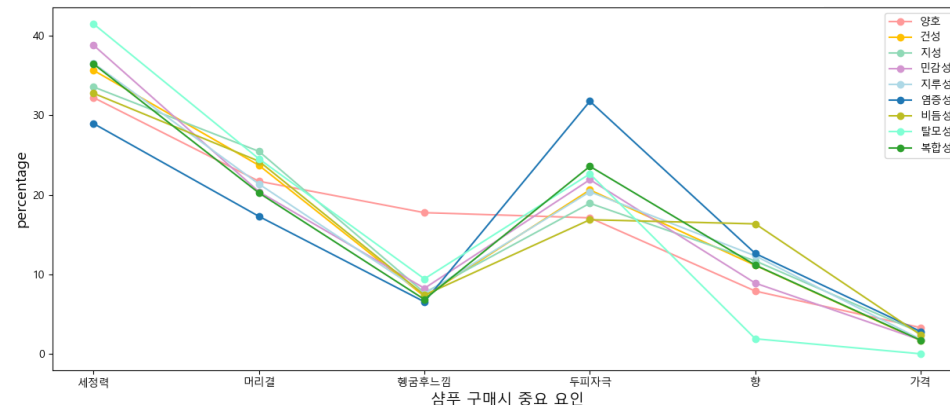
성별 제품 선택 시 중요시하는 요인



연령별 두피유형 분포



두피유형별 중요시하는 샴푸 요인

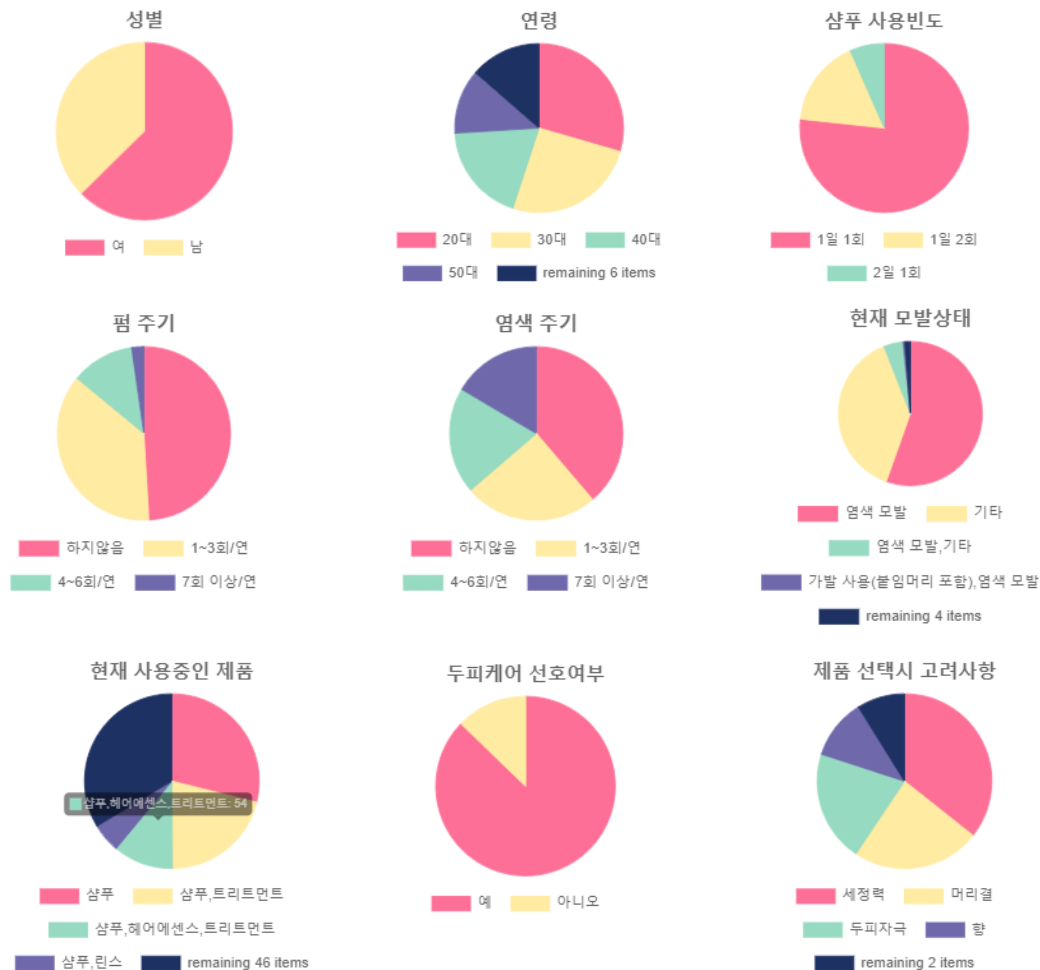




03 메타데이터 : 1) 메타 데이터 기술통계



- ● ● 두피유형별 세부 기술통계 : 9가지 두피유형 각각에 대해 빈도분석 진행
+ 두피유형의 인사이트 정보 제공



빈도분석 시각화 코드

```

+
for idx, col in enumerate(df4.columns):
    idx += 1
    x = df4[col].value_counts() # value_counts로 빈도 확인
    x_topn = x.head(7)

    if len(x) > 7: # 길이가 7을 초과하면 나머지로 통합
        x_topn['remaining (0) items',format(len(x) - 7)] = sum(x[7:]) #
        "remaining 13 items"
        x = x_topn

    plt.subplot(3,3,idx)
    plt.pie(x=x, labels=x.index, autopct='%1f%%', startangle=90,
            counterclock=False,\
            shadow=True)
    plt.title(col)

plt.show()
    
```



03 메타데이터 : 2) 메타 데이터 카이제곱 검정

● ● ● 카이제곱 검정 결과 :

카이제곱 검정 코드

```
columns=['gender', 'age', '샴푸 사용 빈도', '편 주기', '염색 주기 (자가 염색 포함)',
'현재 모발 상태', '현재 사용하고 있는 두피모발용 제품',
'맞춤두피케어 제품사용을 희망(선호)하시나요',
'샴푸 구매시 중요시 고려하는 부분은 무엇인가요', '구분']
```

```
# for 문으로 모든 변수간 카이제곱 검정 실시
```

```
for i in range(9):
    for j in range(9):
        if i != j:
            p = meta_df[columns[i]]
            t=meta_df[columns[j]]
```

```
# 교차표
```

```
ga = pd.crosstab(p,t,margins= False)
print(ga)
```

```
# chi2
```

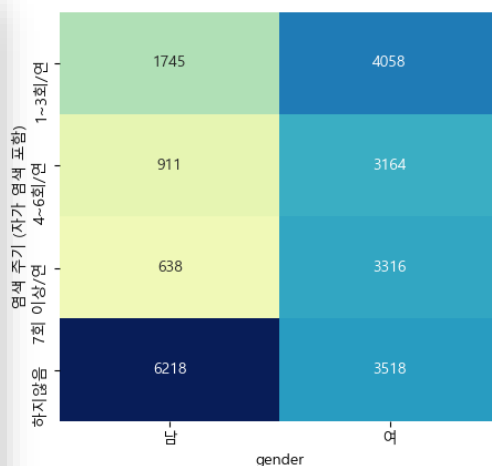
```
chi2, pvalue, dof, expected =
scipy.stats.chi2_contingency(observed=ga,
correction=False)
print("1. 카이제곱 통계량:", chi2)
print("2. p-value:", pvalue)
print("3. df:", dof) # (행의개수-1)*(열의개수-1)
print("4. 기대값 행렬:")
print(pd.DataFrame(expected) )
```

```
# heatmap : 수가 높을수록 색이 짙다.
```

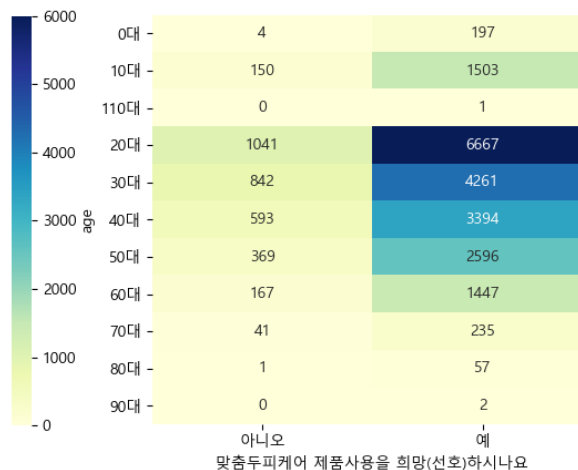
```
ax = sns.heatmap(ga, cmap = "YlGnBu", vmin = 0, vmax =
6000, annot = True,
fmt = '.0f')
plt.title('교차표', fontsize=20)
plt.savefig("교차_"+columns[i]+"_"+columns[j]+".png")
plt.show()
```

```
else:
    print("요거는패스 ~~~")
```

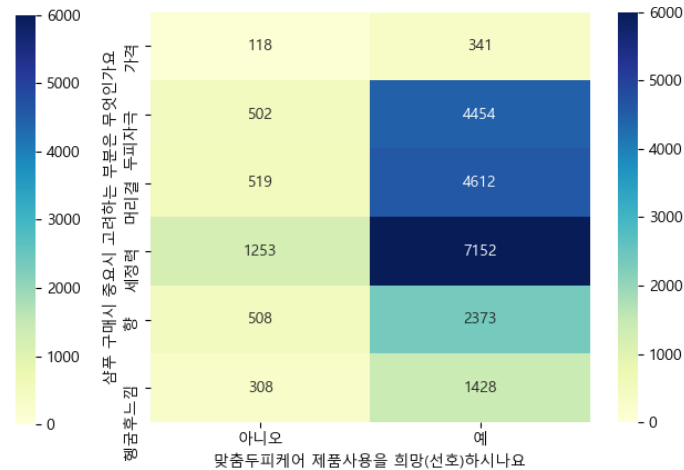
성별에 따른 염색 주기



연령대에 따른 맞춤케어 선호 여부



맞춤케어 선호 여부에 따른 샴푸 선택 요인



남성은 염색 '하지않음'이 높으며
여성은 골고루 분포되어 있음

연령대가 높을수록 맞춤케어를
선호하는 경향이 있음

맞춤케어를 선호하는 그룹은
세정력, 머리결, 두피자극 3가지를
중심으로 다양한 요인을 고려함

맞춤케어를 선호하지 않는 그룹은
세정력 가장 중요시 (feat. 가격)

03 메타데이터 : 3) 메타 데이터 Decision Tree, Naïve Bayes



```
from sklearn.tree import DecisionTreeClassifier
# Decision Tree 코드 # 분류 정확도 : 0.370
model = DecisionTreeClassifier(criterion='entropy', \
    max_depth = 6, min_samples_split = 2, random_state = 1)
parameters = {'max_depth':[1,2,3,4,5,6,7], \
    'min_samples_split':[1,2,3,4,5,6,7], \
    'random_state':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20, \
    21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40, \
    41,42,43,44,45,46,47,48,49,50,123]}
grid_dtree = GridSearchCV(model, param_grid=parameters, \
    cv = 5, refit = True)
grid_dtree.fit(x_train, y_train)
# print(grid_dtree.cv_results_)

score_df = pd.DataFrame(grid_dtree.cv_results_)
print(score_df)

print('GridSearchCV의 최적 파라미터 : ',
    grid_dtree.best_params_)
print('GridSearchCV의 최고 정확도 : ', grid_dtree.best_score_)

# 분류 정확도 확인
y_pred = grid_dtree.predict(x_test)
print('%3f'%accuracy_score(y_test, y_pred))
```

Naïve Bayes 코드 # 분류 정확도 : 0.367

```
+ x 정규화
scaler = MinMaxScaler()
scaler.fit(x)
# 나중에 새로운 값 입력 시 같은 척도로 정규화 하기 위해 fit,
transform 분리
x = scaler.transform(x)
print(x[:3])

# train / test로 분리 ( 7 : 3 )
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size
    = 0.3, \
    random_state = 1) # random_state는 seed와 같음
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
# (16497, 3) (7071, 3) (16497,) (7071,)

from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
print(model)
model.fit(x_train, y_train) # 학습 진행

# 분류 정확도 확인
y_pred = model.predict(x_test)
print('%3f'%accuracy_score(y_test, y_pred))
```

이용 목적 : 분류 예측 모델을 이용해 개인의 설문조사 답변을 바탕으로 두피유형 예측 결과 제공

최종 결정 : 분석 모델 돌린 결과, GridSearchCV를 했음에도 분류 정확도가 낮아 분류예측 서비스 제공 X



03 메타데이터 : 4) Nurdudle 웹 페이지 구현



● ● ● 설문조사를 통해 본인의 두피 상태에 대한 자가 체크 (분류예측 서비스 X)

A Nurdudle Home [두피유형찾기](#) [설문참여하기](#) [통계페이지](#) [건강정보](#)

너의~ 두피 타입이~ 들려 ~

간단한 설문조사를 통해
나와 가장 비슷한 두피타입을 알 수 있습니다.

B Nurdudle Home [두피유형찾기](#) [설문참여하기](#) [통계페이지](#) [건강정보](#)

결과페이지

당신이 선택한

1. 성별 -> 여
2. 연령대 -> 70대
3. 샴푸 사용 빈도 -> 1일 2회
4. 빗 주기 -> 4~6회/연
5. 염색 주기 (자가 염색 포함) -> 1~3회/연
6. 현재 모발 상태 -> 모발이식/사술
7. 두피모발용 제품 -> 헤어에센스
8. 맞춤형두피케어 제품사용을 희망 -> 예
9. 샴푸 구매시 고려 -> 향

당신과 가장 비슷한 타입은 ? 궁금하면 통계정보 누르세요~

[더 자세한 건강정보 클릭 !](#) [더 자세한 통계정보 클릭 !](#)

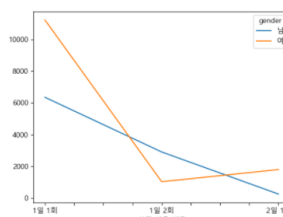
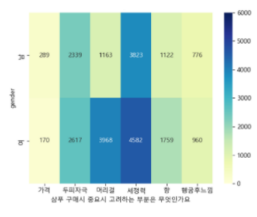
간단한 전체 타입별 남녀_통계 정보

C

당신과 가장 비슷한 타입은 ? 궁금하면 통계정보 누르세요~

[더 자세한 건강정보 클릭 !](#) [더 자세한 통계정보 클릭 !](#)

간단한 전체 타입별 남녀_통계 정보

간단한 두피 타입별 건강정보

[양호](#) [건성](#) [지성](#) [민감성](#) [지루성](#) [염증성](#) [비듬성](#) [탈모성](#) [복합성](#)

양호

특징

- 1) 두피 표면이 맑은 청백색을 띄며 투명도가 높다.
- 2) 각질이 없어 깨끗하여 모공이 완전히 열려 있다.
- 3) 한 개의 모공에 2~3개의 모발이 건강하게 자리잡고 있으며, 모발의 굵기가 일정하다.
- 4) 각질이나 이물질이 없고, 맑은 상태를 유지하며 빈 모공도 거의 보이지 않는다.

건성

특징

- 1) 유 수분이 제거되거나 증발, 건조한 상태. 윤기가 없고 각질이 쌓인다.
- 2) 각화된 표피가 계속 떨어져 나가지 못해 모공을 막고 있는 경우이며, 모발이 가늘어지며 두피 표면이 거칠다.
- 3) 군데군데 각질 변물이 보이며, 모공 주위에 유 수분이 없고 많이 합쳐져 있다.
- 4) 피지막이 없어 손상 받기 쉽고, 염증이 생기기 쉽다.
- 5) 2~3일 머리를 감지 않아도 두피에 기름 피가 확인되지 않는다.

지성

특징

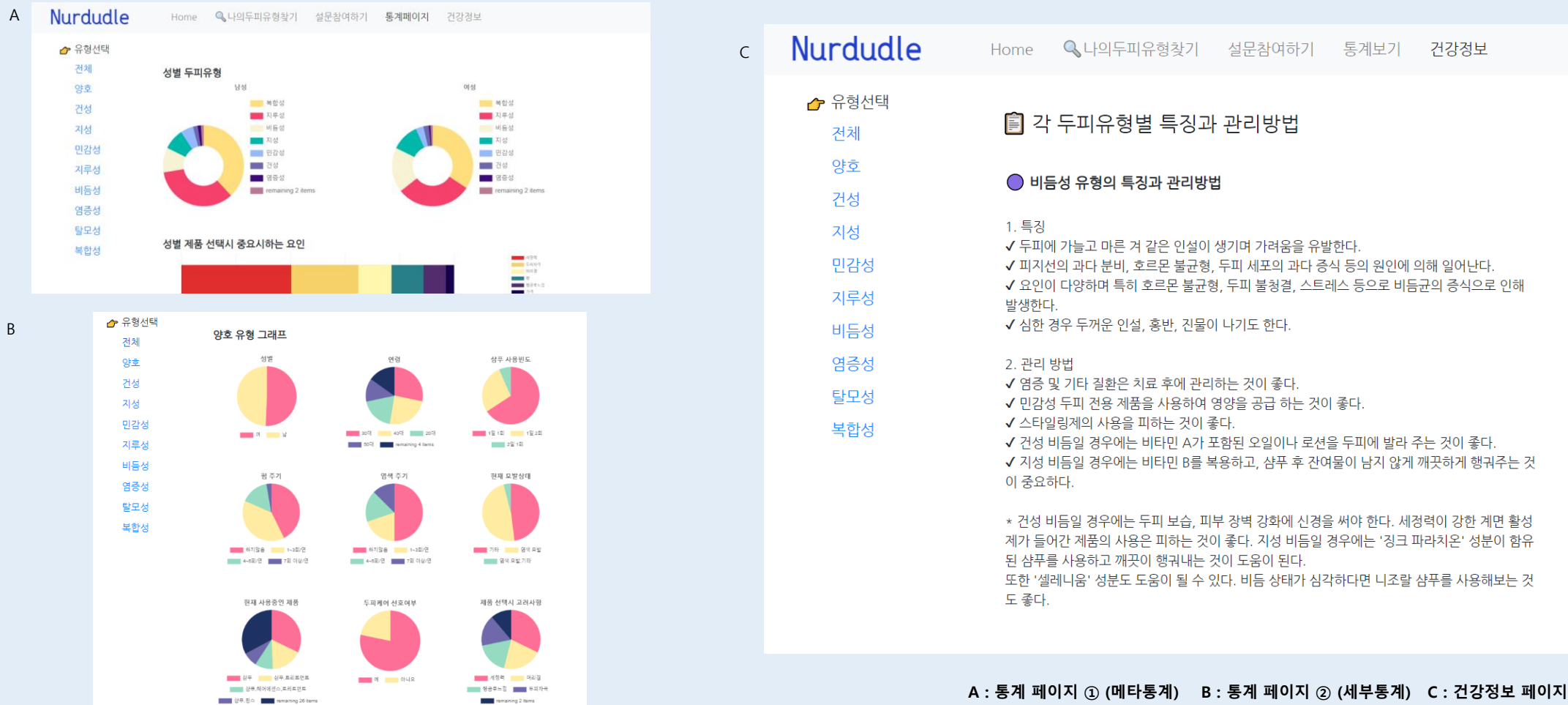
- 1) 과도한 피지 분비로 인해 모공에 물이 고여 있는 것처럼 보인다.
- 2) 두꺼운 각질층과 두피에 투명한 윤기가 없어 특이한 상태이다.
- 3) 비듬과 가려움증을 동반할 가능성이 높다.
- 4) 피지 과다분비로 모발 건조로 피지가 역류하여 지루성 탈모가 발생할 확률이 높다.
- 5) 심한 악취를 동반하여 모발의 탄력도 저하를 유발할 수 있다.



03 메타데이터 : 4) Nurdudle 웹 페이지 구현



통계 페이지와 건강정보 페이지를 통해 다양한 두피 관련 정보 제공



A : 통계 페이지 ① (메타통계) B : 통계 페이지 ② (세부통계) C : 건강정보 페이지

04

04 이미지 분류

1. 두피 이미지 분류 CNN 모델
2. Nurdudle 웹 페이지 구현

04 이미지 분류 : 1) 두피 이미지 분류 CNN 모델링



당신의 두피유형은 무엇입니까 ?

이미지의 라벨링 결과 (ex. "탈모성")을 불러와

Tensorflow의 CNN 딥러닝 알고리즘으로 모델링

CNN 이란? Convolutional Neural Network의 약자로

일반 Deep Neural Network에서 이미지나 영상과 같은 데이터를

처리할 때 발생하는 문제점들을 보완한 방법

● ● ● 이미지 결정 초기 단계

- Aihub에서의 원본 데이터 : **11만개**
(Training Data, Validation Data로 구분)
- 원본 Training Data의 불완전함
& 제한적 개발환경 고려
- Validation Data(원본 데이터의 20%)
사용 후 추가적인 이미지 보강 통해
데이터 양 보완

● ● ● 이미지 결정 최종 단계

- 현재의 개발환경에서 학습 가능한
이미지 데이터 : 10000장
- CNN 학습 시 사용할 원본 데이터 :
9가지 두피유형 X 1100장 = 9900장
- 1100장 이상의 두피유형의 경우:
샘플링으로 1100장 추출
- 1100장 미만의 두피유형의 경우:
이미지 보강으로 1100장 채움

04 이미지 분류 : 1) 두피 이미지 분류 CNN 모델링



두피유형별 개수 파악

```
+
h_t= {'양호':0, '건성':1, '지성':2, '민감성':3, '지루성':4, \
      '염증성':5, '비듬성':6, '탈모성':7, '복합성':8}
## 줄여야 하는 유형 : 8(8453), 4(7483), 6(3462), 2(2344)
## 늘려야 하는 유형 : 3(922), 1(485), 5(214), 0(152), 7(53)
```

이미지 처리 코드

```
+
import os from keras.preprocessing
import image import numpy as np
path = '/content/drive/MyDrive/두피데이터/dataset/features'
def preprocess_img(img_path, target_size=250):
    from keras.preprocessing import image
    img = image.load_img(img_path, target_size=(target_size,
target_size))
    img_tensor = image.img_to_array(img) # scaling into [0, 1]
    img_tensor /= 255.
    return img_tensor

file_list = 라벨링_df["image_file_name"]
# print(file_list)

path='/content/drive/MyDrive/ 두피데이터 /dataset/features'
for i in file_list:
    img_path = os.path.join(path, i)
    x.append(preprocess_img(img_path))
```

두피유형을 1100장으로 줄이는 코드

```
## 1. 줄이기 타입 변수담고 줄이기 -> 합치고 -> 이미지로드 -> 나머지랑 합치기
### 변수담고 줄이기 라벨링_df[라벨링_df["label"]==4][:1100]
```

```
# 줄여야 하는 유형 -> 8,4,6,2
# 제한시켜서 넣는다 -> 나중에 한꺼번에 합칠 예정
복합성= 라벨링_df[라벨링_df["label"]==8][:1100]
지루성= 라벨링_df[라벨링_df["label"]==4][:1100]
비듬성= 라벨링_df[라벨링_df["label"]==6][:1100]
지성= 라벨링_df[라벨링_df["label"]==2][:1100]
```

```
# 줄인 유형들 합치기
줄이기_df = pd.concat([복합성, 지루성, 비듬성, 지성])
```

```
### 줄이기 이미지 로드 후 array화 4400개
file_list = 줄이기_df["image_file_name"]
```

```
x_줄이기 = [] # x_줄이기 -> x_줄이기 타입이미지가 담긴 변수
path = '/content/drive/MyDrive/validation_image'
for i in file_list:
    img_path = os.path.join(path, i)
    x_줄이기.append(preprocess_img(img_path))
```

```
# array 형태로 바꾸기
y_줄이기 = np.array(줄이기_df["label"])
x_줄이기 = np.array(x_줄이기)
print(len(x_줄이기), x_줄이기[0].shape, x_줄이기.shape, x_줄이기[0])
# 4400 (250, 250, 3) (4400, 250, 250, 3) [[0.45882353 0.3725 ...
```



04 이미지 분류 : 1) 두피 이미지 분류 CNN 모델링

두피유형을 1100장으로 늘리는 코드

+

2. 늘리기타입 변수에 담기 → 이미지로드 → 증강 → 합치기 → 나머지랑 합치기
늘리기 컬럼변수에 담기

늘리야 하는 유형들 : 7,0,5,1,3

건성을 예시로 -----

건성 : 485 → 1100로 늘려야 하는 상황

건성 = 라벨링_df[라벨링_df["label"]==1]

file_list = 건성["image_file_name"]

x_건성 = [] # x_건성 : 건성이미지가 담긴 변수

path = '/content/drive/MyDrive/validation_image'

for i in file_list:

img_path = os.path.join(path, i)

x_건성.append(preprocess_img(img_path))

array 형태로 바꾸기

y_건성 = np.array(건성["label"])

x_건성 = np.array(x_건성)

이미지 보강

import numpy as np

from tensorflow.keras.preprocessing.image import ImageDataGenerator

img_gen = ImageDataGenerator(

rotation_range = 10, # 랜덤하게 회전 변형

zoom_range = 0.1, # 확대 축소

shear_range = 0.5, # 축을 기준으로 회전

width_shift_range = 0.1,

horizontal_flip = True, # 좌우반전

vertical_flip = True, # 상하반전

)

건성 이미지 보강

augment_size = 1100-485

randidx = np.random.randint(x_건성.shape[0], size = augment_size)

print("randidx : ", len(randidx))

print(randidx[1])

print(x_건성.shape[0])

y_augment = y_건성[randidx].copy()

x_augment = x_건성[randidx].copy() # 인덱스를 랜덤하게 뽑아내고 복사해서 변수넣는다

x 값만 증식

x_augment = img_gen.flow(

x_augment,

np.zeros(augment_size),

batch_size = augment_size,

shuffle = False).next()[0] # 값만 뽑기 위해

x_건성 = np.concatenate((x_건성, x_augment)) # 기존 485 + 615 = 1100

y_건성 = np.concatenate((y_건성, y_augment))

이미지 증강 후 합치기

y_늘리기 = np.concatenate((y_탈모성, y_민감성, y_건성, y_양호, y_염증성))

x_늘리기 = np.concatenate((x_탈모성, x_민감성, x_건성, x_양호, x_염증성))

3. x_늘리기 & x_줄이기 합치기

x = np.concatenate((x_늘리기, x_줄이기))

y = np.concatenate((y_늘리기, y_줄이기))



04 이미지 분류 : 1) 두피 이미지 분류 CNN 모델링



전이학습 전 모델 # 분류 정확도 : 0.389

```
+ from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D,
Dropout, Flatten

n_classes = 9

model = Sequential()
#conv1
model.add(Conv2D(128, kernel_size=(3,3), input_shape=(250,250,3),
padding='same',
activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))

#conv2
model.add(Conv2D(64, kernel_size=(3,3), padding='same',
activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))

#Flatten
model.add(Flatten())

#Dense
model.add(Dense(128, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(n_classes, activation='softmax'))
```

전이학습 후 모델 1 - VGG # 분류 정확도 : 0.542

```
+ from tensorflow.python.keras.applications.vgg16 import VGG16
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Dropout,
Flatten

n_classes = 9 # 복합성포함

transfer_model = VGG16(weights='imagenet', include_top=False,
input_shape=(250, 250, 3))
transfer_model.trainable=False
model = Sequential([
    transfer_model,
    Conv2D(256, kernel_size=(3,3), input_shape=(250,250,3),
padding='same',
activation='relu'),
    MaxPool2D(pool_size=(2,2)),
    Dropout(0.5),

    Flatten(),
    Dense(512, activation='relu'),
    Dense(256, activation='relu'),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(n_classes, 'softmax')
])
```

04 이미지 분류 : 1) 두피 이미지 분류 CNN 모델링



전이학습 후 모델 2 - MobileNet # 분류 정확도 : 0.623

```
+ from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Dropout, Flatten

n_classes = 9 # 복합성포함

transfer_model = MobileNetV2(input_shape=(250, 250, 3),
include_top=False, weights='imagenet')
transfer_model.trainable=False

model = Sequential([
    transfer_model,
    Conv2D(256, kernel_size=(3,3), input_shape=(250,250,3),
padding='same',
    activation='relu'),
    MaxPool2D(pool_size=(2,2)),
    Dropout(0.5),

    Flatten(),
    Dense(512, activation='relu'),
    Dense(256, activation='relu'),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(n_classes, 'softmax')
])
```

모델 학습 코드

```
+ from tensorflow.keras.optimizers import Adam

optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,
loss='sparse_categorical_crossentropy', metrics=['acc'])

# EarlyStopping
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
earlystop = EarlyStopping(monitor='val_loss', mode='auto', patience=3)

history = model.fit(x_train, y_train, epochs=100, batch_size=64,
verbose=1,\
    callbacks=[earlystop], validation_split=0.2)

print(model.evaluate(x_test,y_test))
```

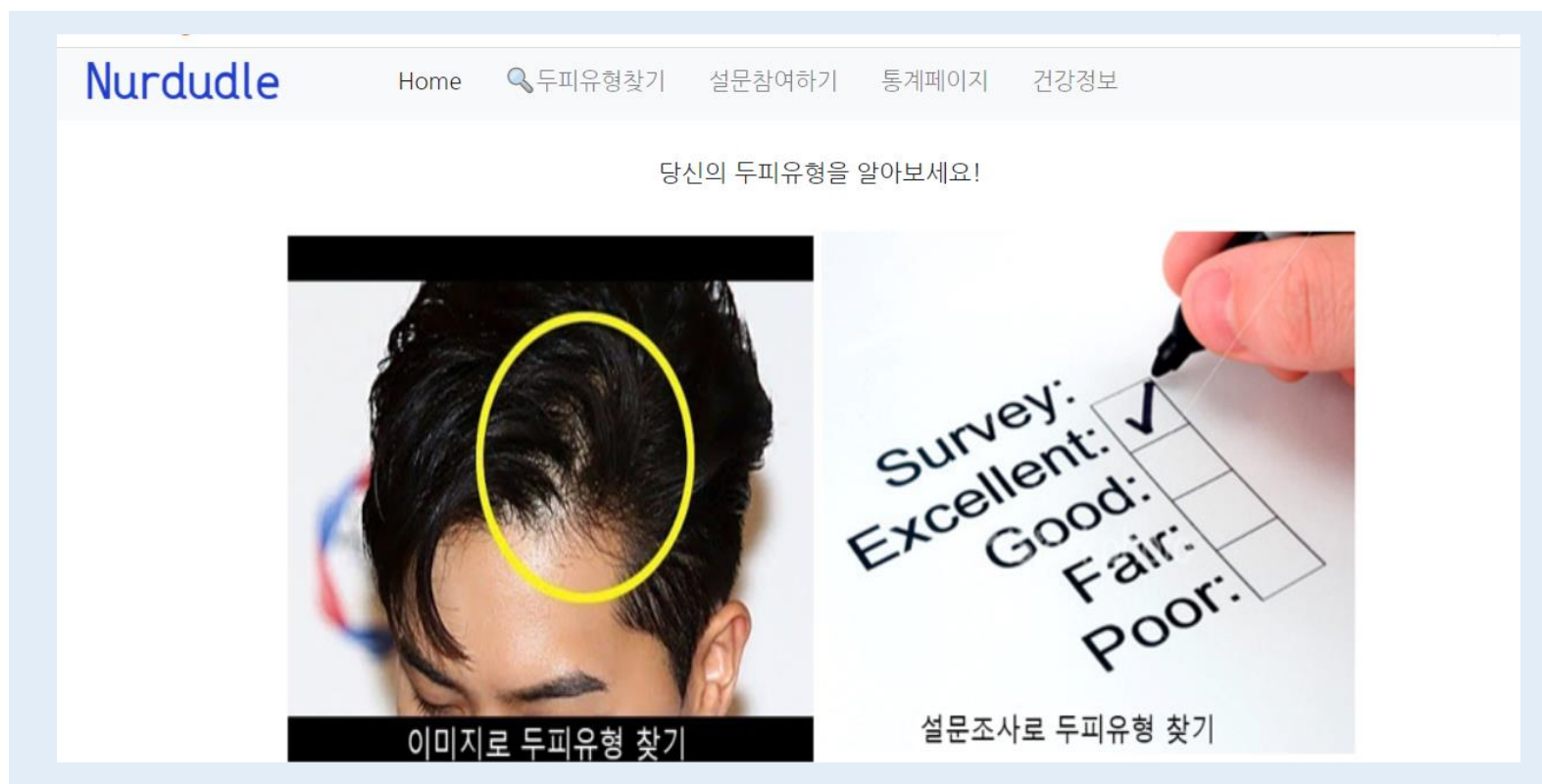
- 가장 높은 분류 정확도가 나온 [전이학습 후 모델 2 - MobileNet]
최종 이미지 분류 모델로 확정

- ✓ Dense Layer를 쌓을수록 분류 정확도가 올랐으나, 일정 수준을 넘어서면 학습 속도가 느려지며, 정확도에 큰 차이가 발생하지 않았음
- ✓ Adam의 learning_rate로 0.001일 설정했을 때 결과가 가장 좋았음
- ✓ 학습 시 batch_size 또한 중요한 영향 요인이었으며, 64, 128일 때 가장 적합했음

04 이미지 분류 : 2) Nurdudle 웹 페이지 구현



- ● ● Nurdudle 메인 페이지에서 [이미지로 두피유형 찾기], [설문조사로 두피유형 찾기] 선택





04 이미지 분류 : 2) Nurdudle 웹 페이지 구현



● ● ● [두피유형 찾기] 페이지

A Nurdudle Home 두피유형찾기 설문참여하기 통계페이지 건강정보

이미지 불러오기

파일 선택 선택된 파일 없음

[결과보기](#)

B Nurdudle Home 두피유형찾기 설문참여하기 통계페이지 건강정보

이미지 불러오기

파일 선택 0013_A2LE011_63457_4_14.jpg



[결과보기](#)

A : 이미지 업로드 전 B : 이미지 업로드 직후



04 이미지 분류 : 2) Nurdudle 웹 페이지 구현



● ● ●

[두피유형 찾기]

결과 페이지

Nurdudle
Home
두피유형찾기
설문참여하기
통계페이지
건강정보

고객님의 두피유형은 복합성입니다.

[복합성의 건강정보]
더보기

특징

위에 해당하는 유전형 해당 증상이 복합적으로 나타날 경우, 병원에서 상담을 받는 것이 좋다.

두피 건강에 영향을 끼치는 요인은 매우 다양하기 때문에 관리도 복잡할 수도 있지만 어느정도 증상이 발현된 뒤에는 치료가 늦을수록 호전이 어려워지는 경우가 많기 때문에 1주일 이상 증상을 겪었다면 초기 단계에 진료를 받아 건강한 두피와 모발을 꾸준히 유지하는 것이 가장 좋다.

[복합성의 통계그래프]
더보기

성별

연령

[복합성의 제품추천]

두피유형 분류 예측 결과 출력

두피 특징 및 관리 방법 안내

두피유형 맞춤 헤어 제품 추천

건강정보 페이지 /
통계 페이지 보러가기

Slide 26



05 시연

●●● 06 기대효과 및 한계점

●●● 기대효과 :

개인 이용자 측면

이미지 업로드라는 간편한 방식으로 본인 두피유형 파악 가능

기업 이용자 측면

협업 기업의 두피모발용 제품에 대해
더욱 상세한 추천 서비스를 제공함으로써 [Nurdudle](#) 서비스의 상업성 보장

●●● 한계점 :

장비 측면

전문 두피 진단기기에 대해 개인의 접근성 낮음

서비스 측면

전문 두피 진단기기로 촬영된 두피 이미지로 이미지 분류가 이뤄지고 있어,
개개인의 현 두피 상태에 대한 진단 X

개발환경 측면

개발환경으로 인해 모델 학습 시 들어가는 이미지 데이터 양이 제한적



100일 800시간 동안 다들 고생 많으셨습니다.