

기업 프로젝트 with (주)여보야

정성·정량적 요소를 결합한 배우자 추천 시스템

U&I (박재운, 황선진, 강다빈, 전해민)



목차

Contents

- | | |
|----|--------------------------|
| 01 | 프로젝트 주제 |
| 02 | 서비스 시연 예시 |
| 03 | 프로젝트 진행 배경 및 목적 |
| 04 | 서비스 타겟 설정 및 혜택 |
| 05 | 프로젝트 개발환경 |
| 06 | 역할분담 및 개발 일정 |
| 07 | 서비스 프로세스 및 기술소개 |
| 08 | 기대효과 |
| 09 | 프로젝트 자체평가 및 참고자료, 깃허브 주소 |
| 10 | 추후 프로젝트 제안서 |

01:프로젝트 주제

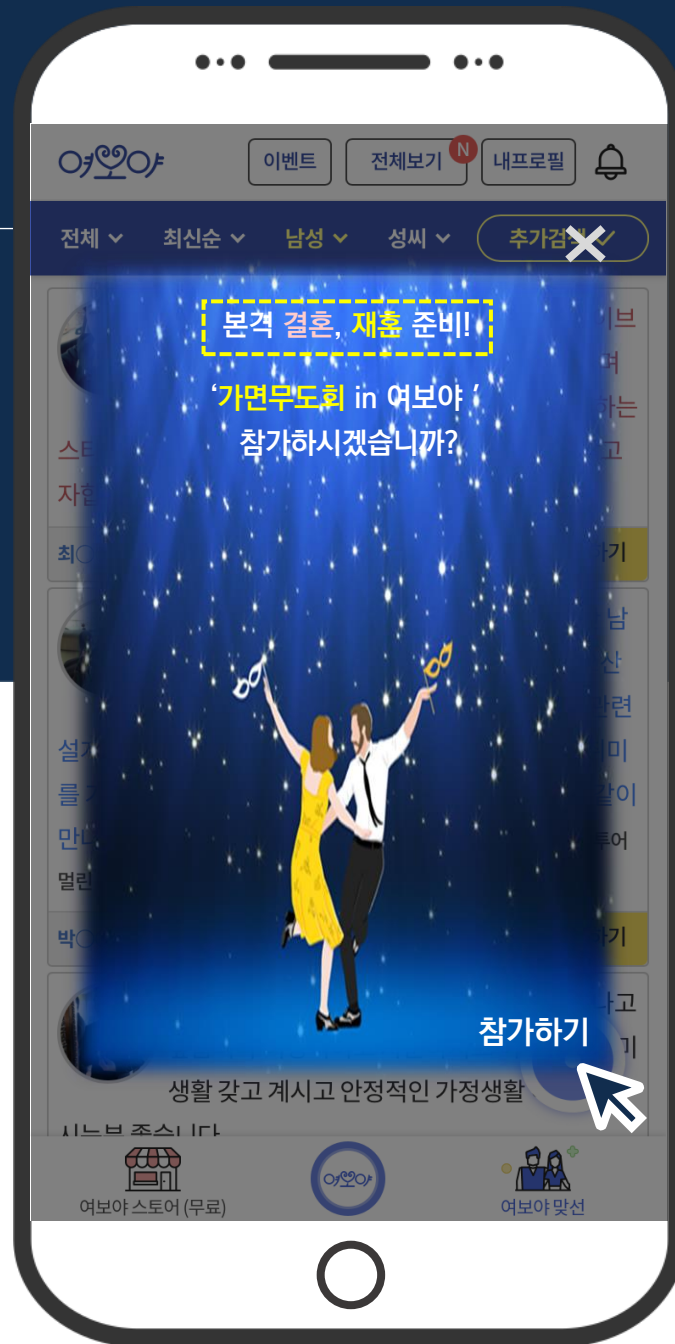
‘동물 가면무도회 in 여보야’에 참가하시겠습니까?



02:서비스 시연 예시



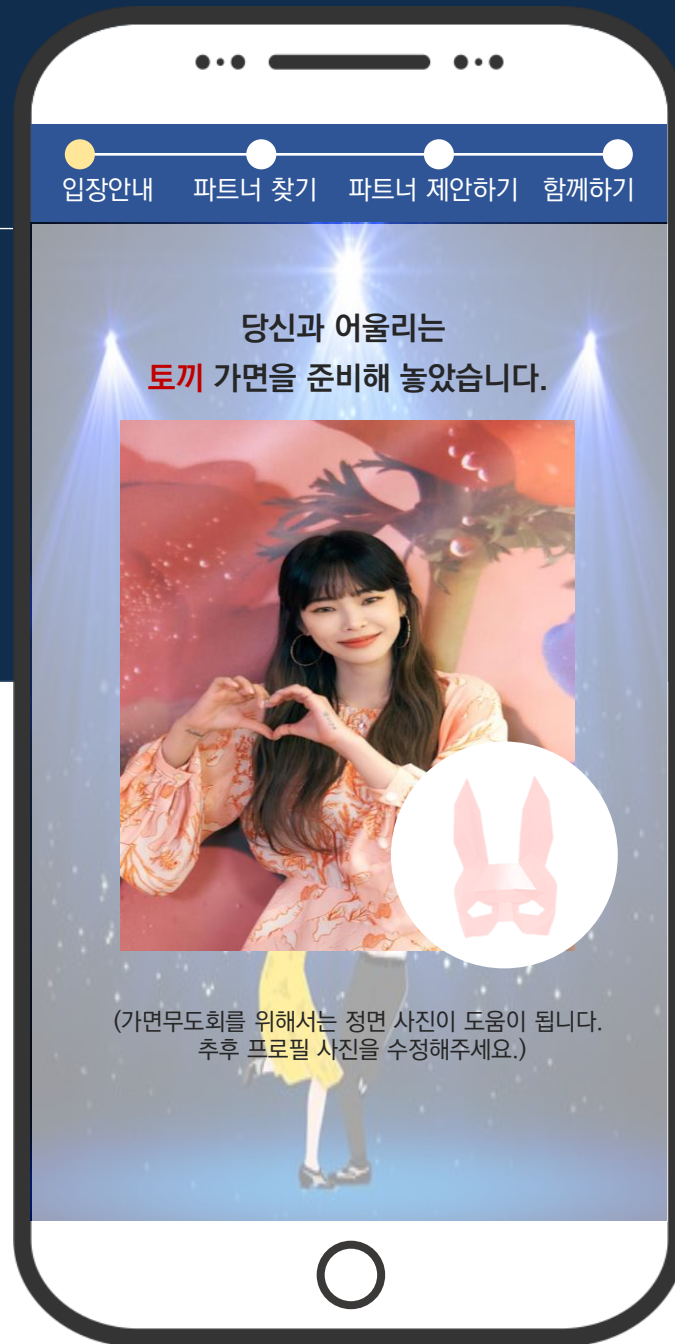
“여보야 앱 접속시 ‘가면무도회 in 여보야’ 팝업창 클릭”



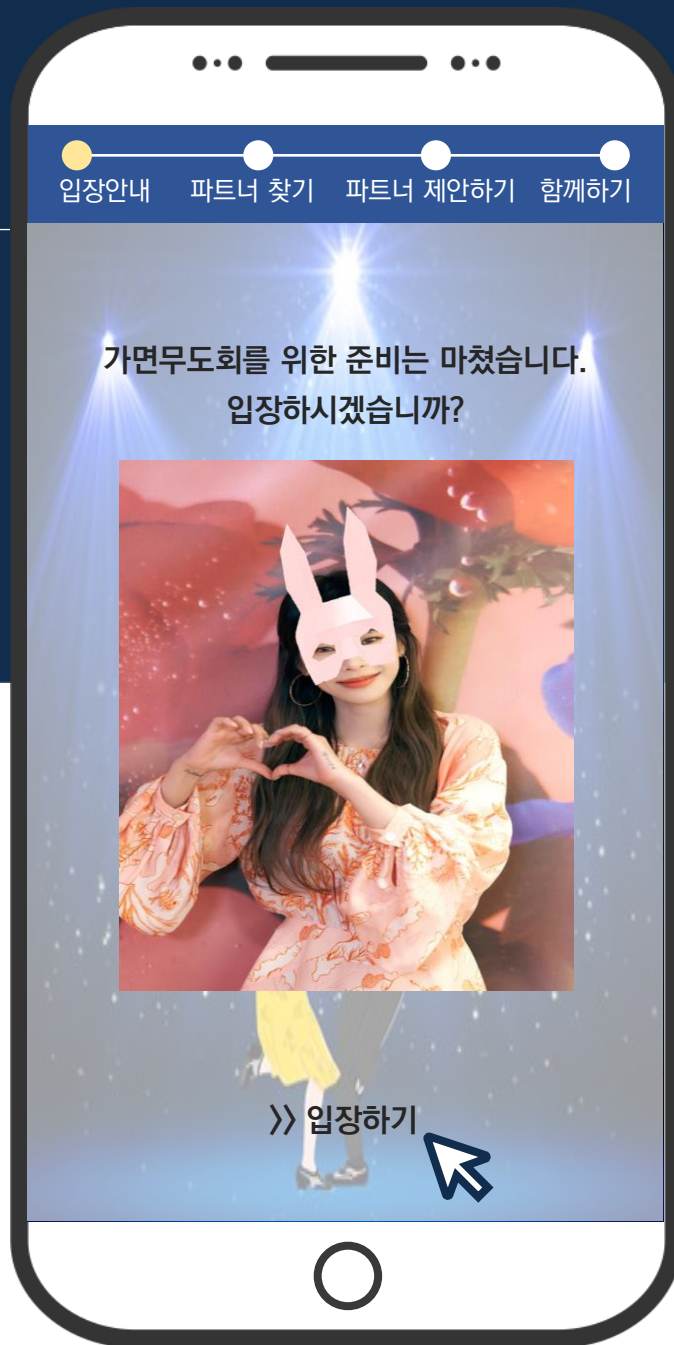
02:서비스 시연 예시



“여성 유저를 기준으로 프로세스 진행”
“동물상 분류에 따라 해당 유저의 가면이 결정됨”



02:서비스 시연 예시



02:서비스 시연 예시



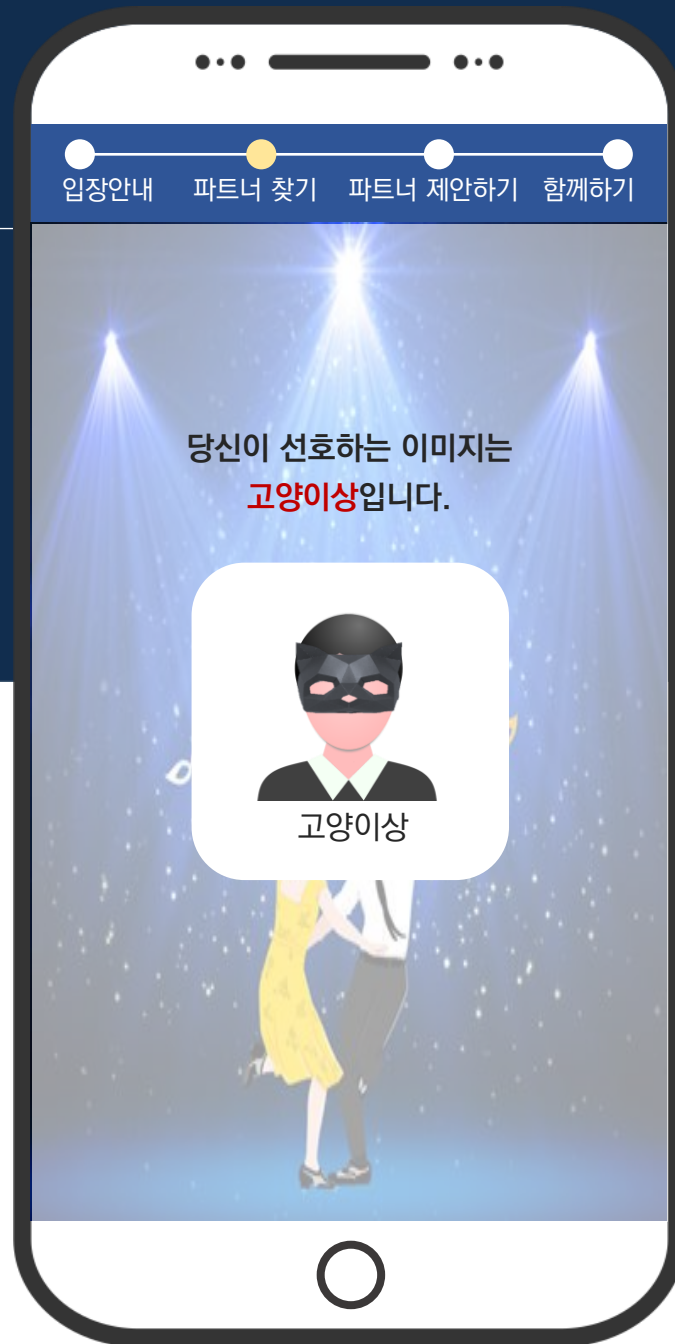
“남성 유저의 대표 동물상 4가지 중에서
여성 유저가 **파트너**로서 선호하는 이미지 택 1”



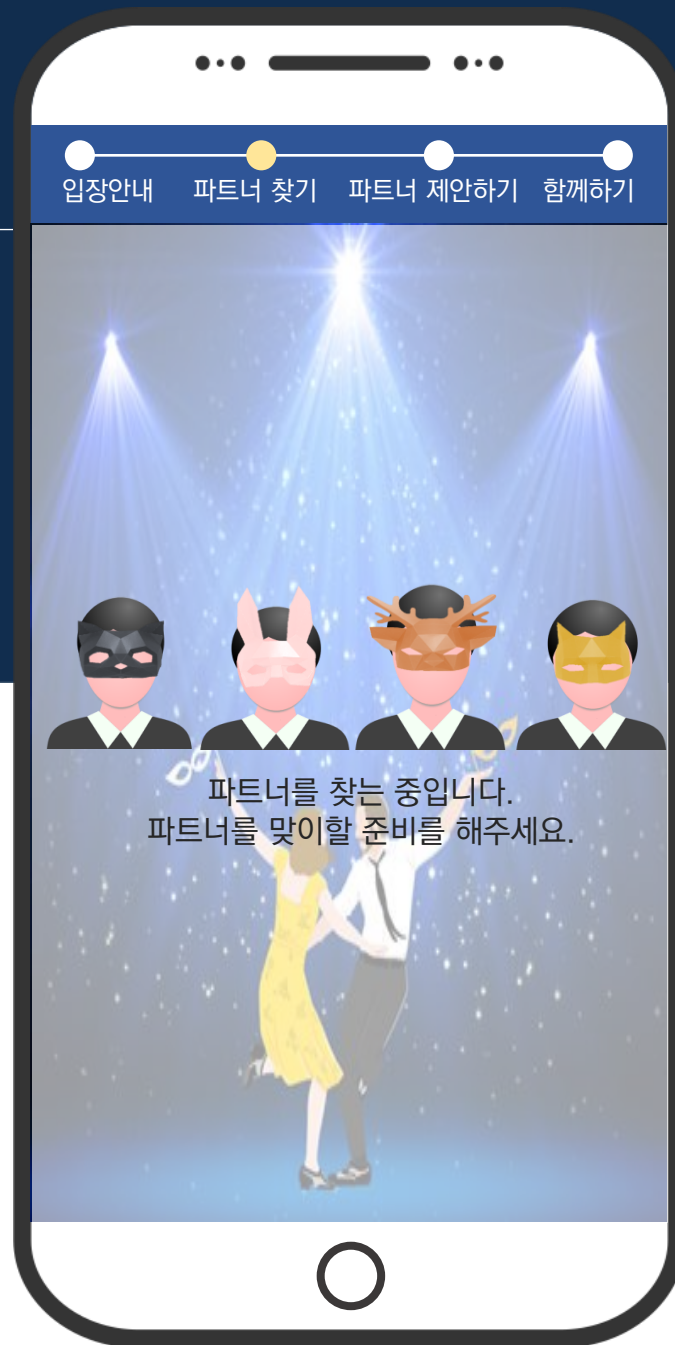
02:서비스 시연 예시



“선호하는 이미지 다시 한번 보여줌”



02:서비스 시연 예시



02:서비스 시연 예시

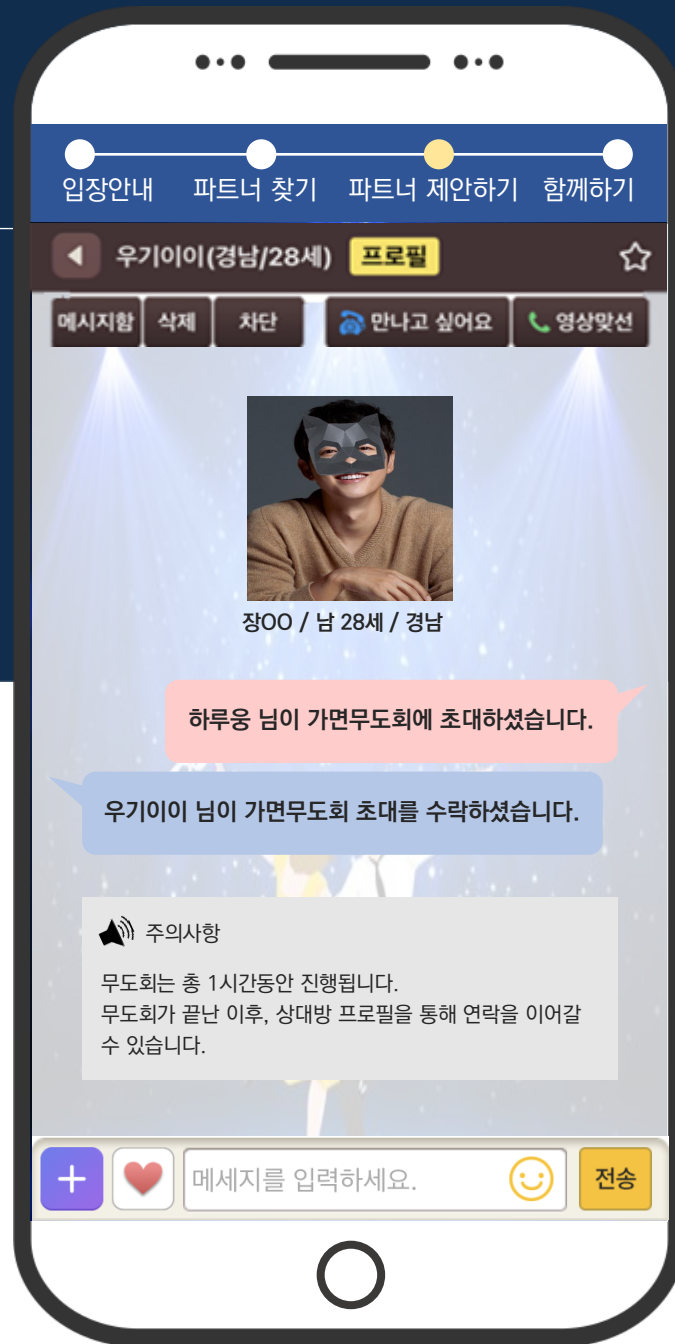
“유저가 선호하는 동물상 이미지를 가진 **파트너 TOP3**가
가면이 합성된 프로필 이미지와 함께 추천됨”



02:서비스 시연 예시



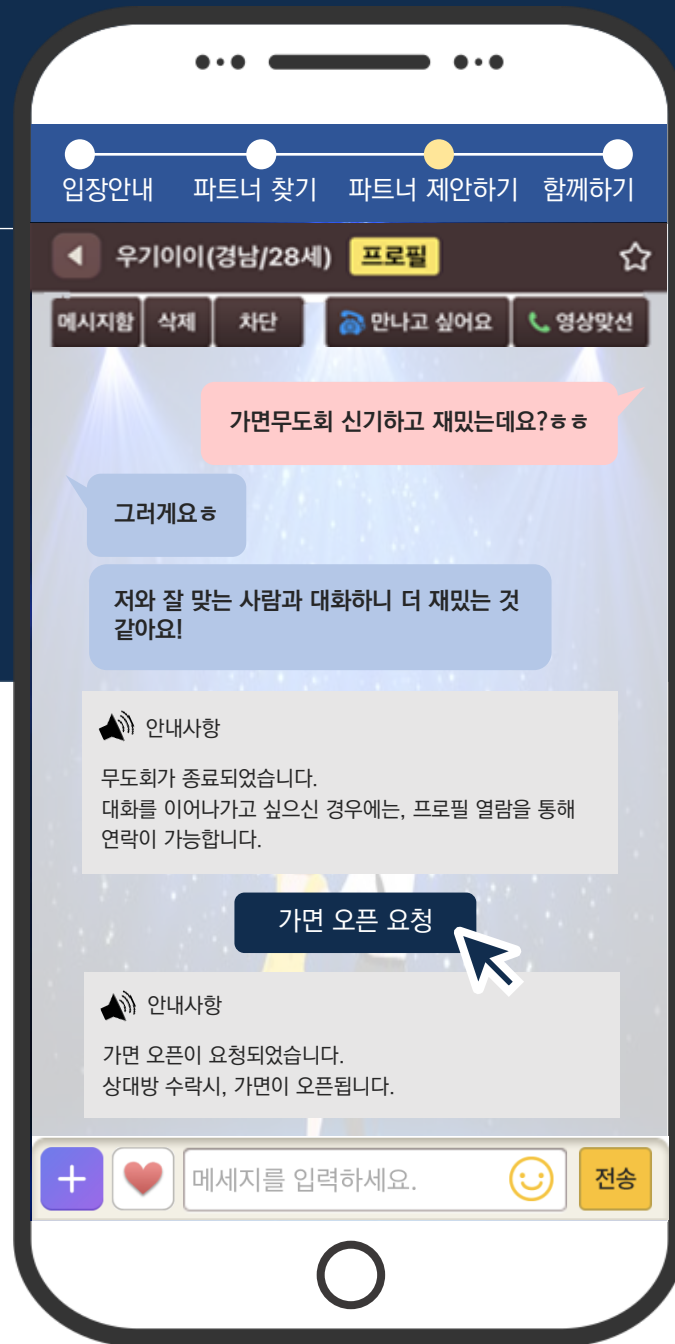
“파트너의 응답에 따라 대화의 진행 여부 결정됨”
“가면무도회 속 대화는 오직 1시간 동안만 가능”



02:서비스 시연 예시



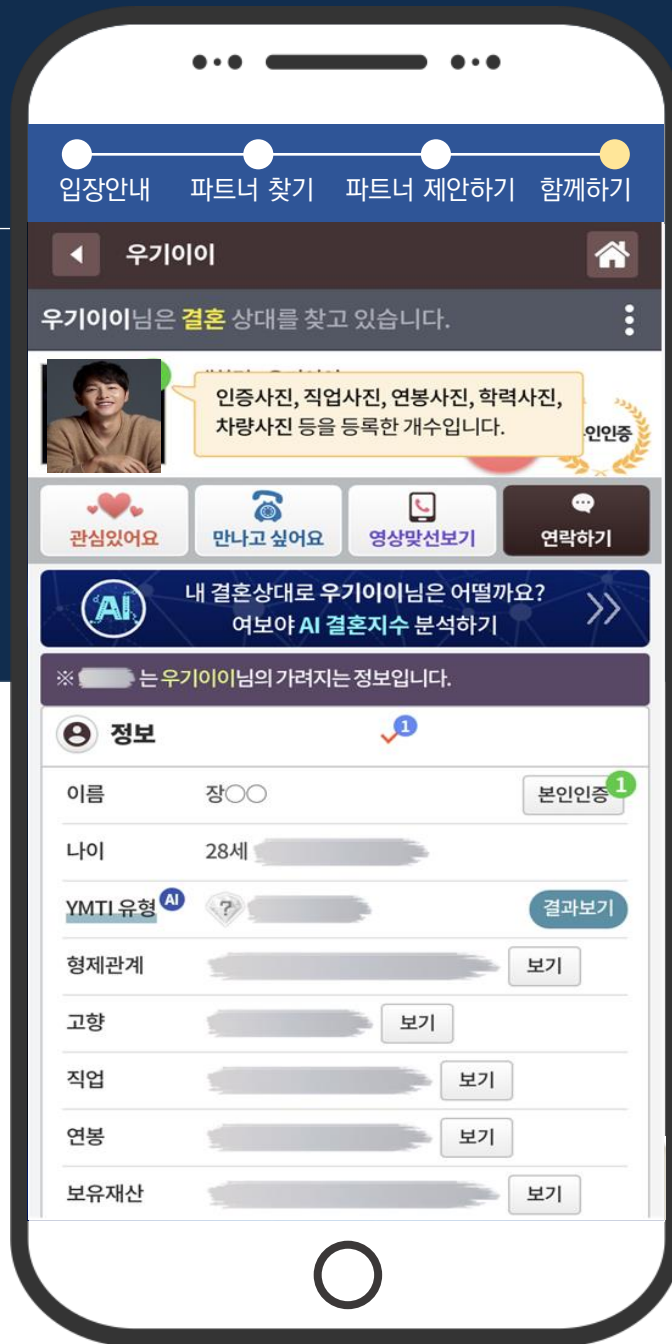
“가면무도회의 제한시간이 지난 이후에는
프로필 열람 / 가면 오픈 요청으로 대화를 이어나갈 수 있음”



02:서비스 시연 예시



“프로필 열람 버튼 클릭시 화면”



03:프로젝트 진행 배경 및 목적

1

기업의 니즈

- 신선한 아이디어 기반의 배우자 추천 시스템

2

EDA(탐색적 데이터 분석) 결과

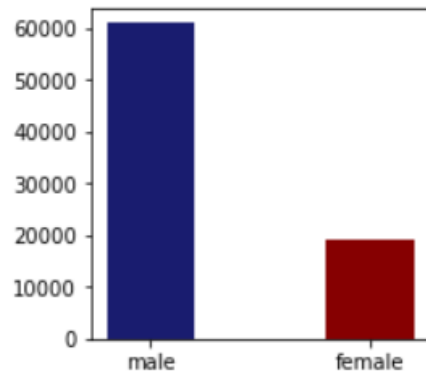
- 여성 유저의 수, 활동량이 적음
- 서비스 활성화를 위해 남, 녀 비율을 맞출 필요가 있음
- 여성 유저의 유입이 필요하다고 판단함

3

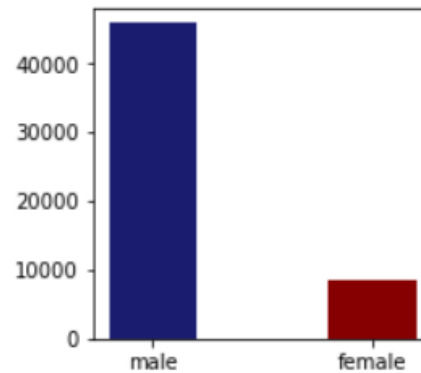
인사이트 도출

- 이미지 활용
- 얼굴 노출 부담을 최소화 할 수 있는 서비스

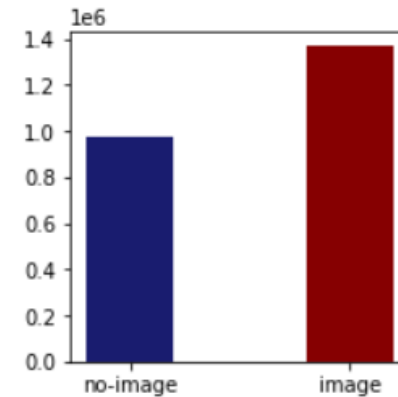
〈 남녀 유저 비율 〉



〈 남녀 이미지 비율 〉



〈 여성 활동량 비율 〉



04:서비스 타겟 설정 및 혜택



TARGET

- Main : 여성 유저
- Sub : 서비스 이용량/활동량이 적은 유저

BENEFIT

- ‘가면무도회’라는 컨셉을 통해 얼굴 노출의 부담감 감소
- 적은 부담감 덕분에 서비스 이용 활성화 가능
- 활동량 증가로 인해 다양한 만남의 기회 증가

05:프로젝트 개발환경



소통

- Slack
- Google Meet
- Kakao Talk
- Nate On



개발환경

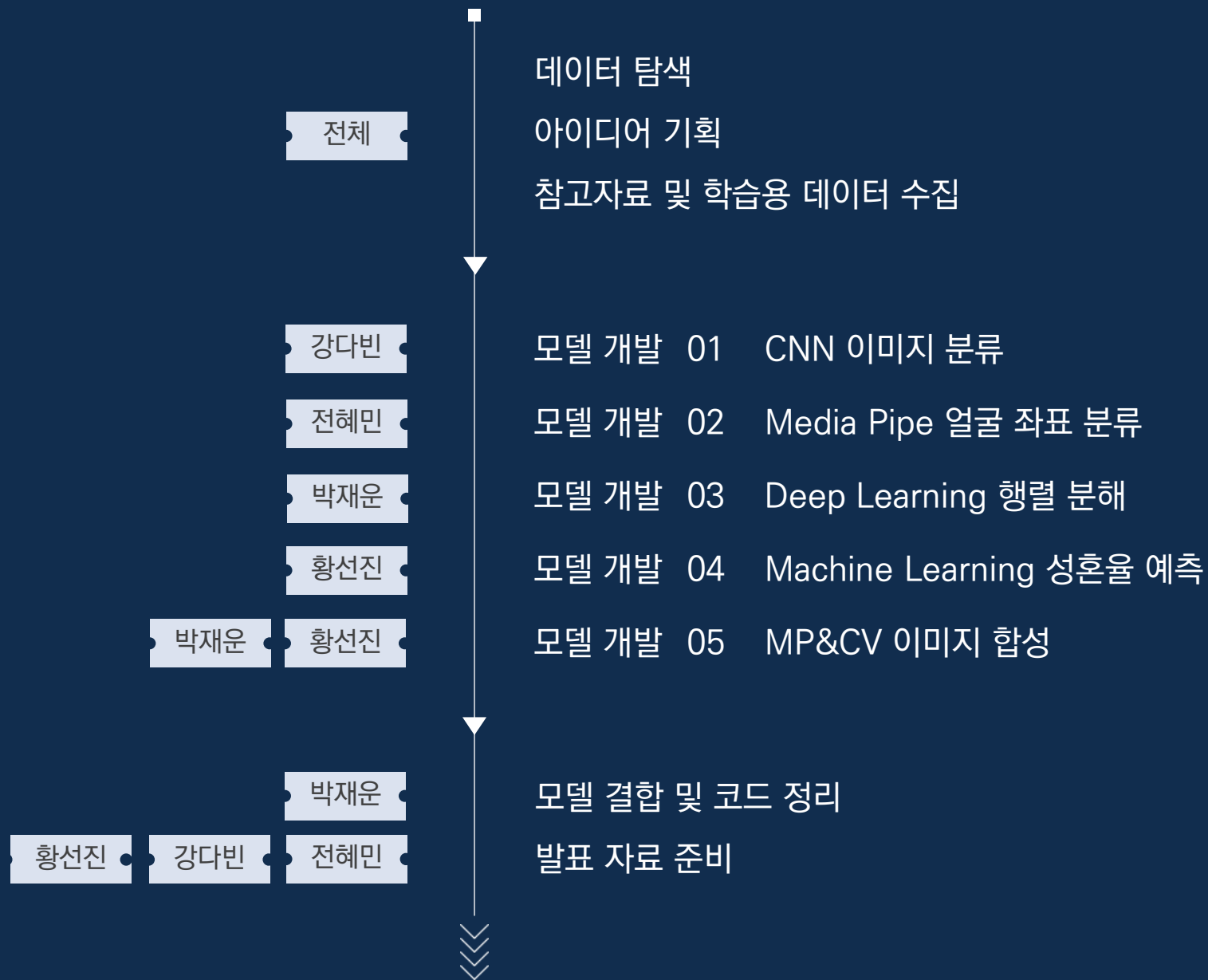
- Anaconda
- Google Colab



코드 및 문서 관리

- GitHub
- Google Meet

06: 역할 분담 및 개발 일정

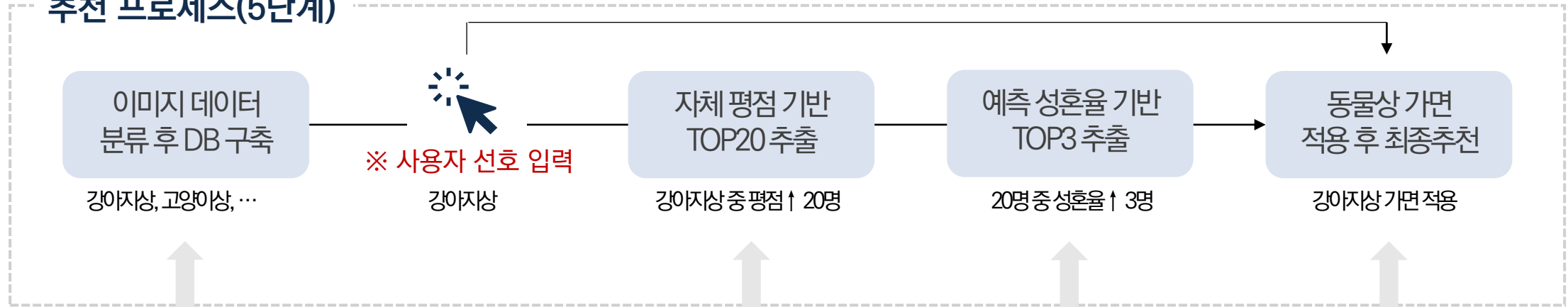


07:서비스 프로세스 및 기술소개

전체 프로세스



추천 프로세스(5단계)



적용 모델 및 AI기술 (5가지)



정성적 추천

정량적 추천

정성 + 정량적 추천

07:서비스 프로세스 및 기술소개

세부 프로세스 1단계 <이미지 데이터 분류 후 DB 구축>



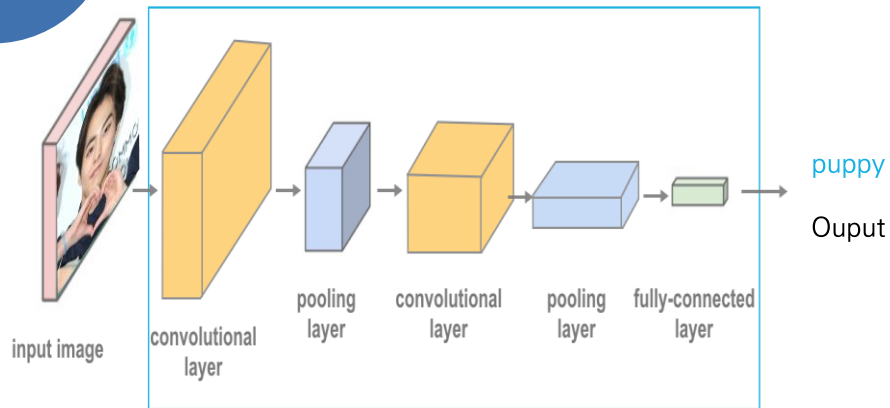
[입력 데이터] 여보야 유저 이미지 → 분류모델 → [결과 데이터] 가장 유사한 동물상(ex: puppy) → DB구축

(학습 데이터) 연예인 이미지 - 웹 크롤링



CNN
이미지
분류

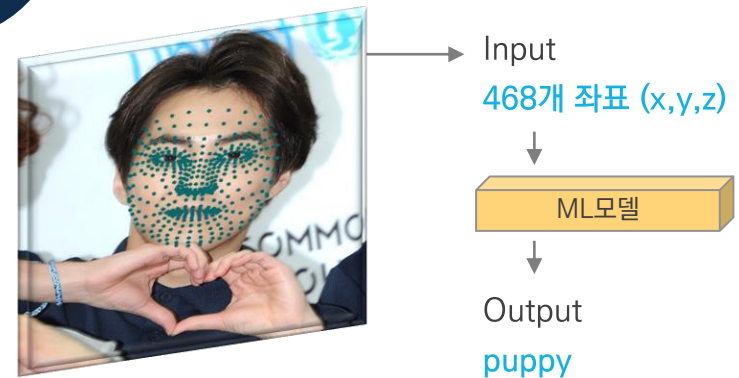
연예인 이미지(픽셀) 학습



CNN

Media
Pipe
얼굴 좌표
분류

연예인 얼굴 좌표 학습



[참고] 관련 코드(요약)

- ✓ CNN 모델구축(convolution layer 2계층, fully connected layer 1계층)
- ✓ 총 학습 파라미터 수 : 1,625,797개

```
from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Dropout
from tensorflow.keras.models import Sequential

# 모델 구축
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=x_train.shape[1:]),
    MaxPooling2D(2,2),
    Dropout(0.5),
    Conv2D(64, (3,3), activation = 'relu'),
    MaxPooling2D(2,2),
    Dropout(0.5),
    Flatten(),
    Dense(128, activation = 'relu'),
    Dropout(0.5),
    Dense(nb_classes, activation = 'softmax')
])

# 모델 컴파일
model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['acc'])

# 모델 학습
history = model.fit(x_train, y_train,
                   validation_data=(x_test, y_test),
                   batch_size=32,
                   epochs=100,
                   verbose=0,
                   callbacks=[checkpoint])
```

- ✓ Mediapipe Facemesh landmark를 활용, 이미지에서 좌표추출(468개 * [x, y, z])
- ✓ 해당 좌표들을 데이터프레임화하여 머신러닝 학습

```
# 좌표 추출
import mediapipe as mp
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(static_image_mode=True)

image = cv2.imread(f_folder_path + '/' + file)
rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
result = face_mesh.process(rgb_image)

# 데이터 프레임 생성
temp = ['class']
for i in range(1,469):
    temp += ['x{}'.format(i), 'y{}'.format(i), 'z{}'.format(i)]

for facial_landmarks in result.multi_face_landmarks:
    for i in range(0,468):
        pt = facial_landmarks.landmark[i]
        x = pt.x
        y = pt.y
        z = pt.z
        face += [x,y,z]

# 머신러닝 학습
from sklearn.linear_model import Perceptron
ppn = Perceptron(max_iter=40, eta=0.1, tol=1e-3, random_state=1)
ppn.fit(x_train_std, df_y_train)
y_pred = ppn.predict(x_test_std)
```

07:서비스 프로세스 및 기술소개

세부 프로세스 2단계 <예측 평점 TOP20 추출>

[입력 데이터] 유저 ID & 선호 동물상 → 행렬분해 모델 → [결과 데이터] 선호 동물상 중 평점 TOP20 이성 ID

(활용 데이터) 사용자 프로필 열람 / 관심 / 연락 로그 데이터

👤 프로필 열람 (1점) ❤️ 관심있어요 (2점) 📞 연락(3점) > 유저 - 이성 유저간 자체 평점을 계산

Deep
learning
행렬
분해

(유저 - 이성 유저) 자체평점 행렬을 기반으로 예측 자체평점 행렬 생성

※ TensorFlow를 활용하여 직접 모델링

	이성1	이성2	이성3	이성4
유저A	3	2		
유저B		1	5	4
유저C	3	1		3
유저D				4



	이성1	이성2	이성3	이성4
유저A	2.99	2.11	4.71	2.92
유저B	2.99	0.98	5.08	3.89
유저C	3.01	1.12	4.89	3.09
유저D	2.77	1.21	4.65	4.21

[참고] 관련 코드(요약)

- ✓ 로그 데이터(view, concern, call) 기반으로 자체 평점을 계산
- ✓ (유저 - 여성 유저) 자체 평점 행렬 생성

```
import pandas as pd
import numpy as np

# 자체 평점 계산 * 여성기준 평점 : ((view*1) + (concern*1.5) + (call*2))

female_view = view[(view['mem_sex'] == 'f') & (view['ptr_mem_sex'] == 'm')]
female_view['view_weight'] = 1
female_view = female_view[['mem_no', 'ptr_mem_no', 'view_weight']]

female_concern = concern[(concern['mem_sex'] == 'f') & (concern['ptr_mem_sex'] == 'm')]
female_concern['concern_weight'] = 1.5
female_concern = female_concern[['mem_no', 'ptr_mem_no', 'concern_weight']]

female_call = call[(call['mem_sex'] == 'f') & (call['ptr_mem_sex'] == 'm')]
female_call['call_weight'] = 2.0
female_call = female_call[['mem_no', 'ptr_mem_no', 'call_weight']]

# 평점 행렬 생성
merge1 = pd.merge(male_call, male_concern, how='right', on=['mem_no', 'ptr_mem_no'])
merge2 = pd.merge(merge1, male_view, how='right', on=['mem_no', 'ptr_mem_no'])

df_rating['view_weight'].fillna(0, inplace=True)
df_rating['concern_weight'].fillna(0, inplace=True)
df_rating['call_weight'].fillna(0, inplace=True)
df_rating['rating'] = df_rating['view_weight'] + df_rating['concern_weight']
                    + df_rating['call_weight']
```

- ✓ (딥러닝) TensorFlow 행렬분해 모델링(Function API 활용) 및 학습

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Embedding, Dot, Add, Flatten

# TensorFlow 모델링 - Functional API 활용
u_in = Input(shape=(1,), name = "Input_User")
m_in = Input(shape=(1,), name = "Input_Movie")
u_mat = Embedding(N, n_factors, name="U_mat")(u_in)
m_mat = Embedding(M, n_factors, name="M_mat")(m_in)
x = Dot(axes=2, name="Dot_Product")([u_mat, m_mat])

u_bias = Embedding(N, 1, name="u_bias")(u_in)
m_bias = Embedding(M, 1, name="m_bias")(m_in)
x = Add(name="Biases")([x, u_bias, m_bias])

x = Flatten(name="Flatten")(x)
my_model = Model(inputs=[u_in, m_in], outputs=x)

# 학습
n_epochs = 20
batch_size = 256
learn_rate = 0.0001

my_model.compile(loss = "mse", metrics=["mse"])
my_summary = my_model.fit([df_rating['user'].values, df_rating['item'].values],
                          df_rating['rating'].values - mu, epochs=n_epochs,
                          batch_size=batch_size, validation_split= 0.3, verbose= 1)
```


07:서비스 프로세스 및 기술소개

세부 프로세스 3단계 <예측 성혼율 기반 TOP3 추출>

[입력 데이터] 유저 ID & TOP20 이성 ID → 머신러닝 모델 → [결과 데이터] 예측 성혼율 TOP3 이성 ID

(활용 데이터) 여보야 성혼 데이터 및 고객 전체 데이터



여보야 성혼 데이터 819쌍



미성혼 데이터 (임의추출) 819쌍

Machine
learning
성혼율
예측

성혼(1) / 미성혼(0) 데이터 - 머신러닝 학습

Input

사용자 프로필 정보

이성 프로필 정보

데이터 전처리

수치형 변수(5개)

나이 차이, 키 차이, 연봉 차이, 자산 차이, 학업 차이

범주형 변수(각 12개)

초혼/재혼, 지역(10개 권역)

ML모델 (로지스틱회귀)

Output 성혼율 예측 (0~1)

[참고] 관련 코드(요약)

- ✓ 성혼 데이터의 박스플롯의 upper fence를 적용하여 미성혼 샘플을 추출
- ✓ 초혼(남) / 초혼(여) / 재혼(남) / 재혼(여)를 구분하여 임의 매칭

```
import pandas as pd
import numpy as np

# 미성혼 샘플 추출
def upper_fence(series) :

    Q1 = series.quantile(0.25)
    Q3 = series.quantile(0.75)
    IQR = Q3 - Q1
    Upper_Fence = Q3 + (2 * IQR)

    return Upper_Fence

w_m_span = upper_fence(married[married['mate_slct'] == 'w']['span'])
r_m_span = upper_fence(married[married['mate_slct'] == 'r']['span'])
w_f_span = upper_fence(married[married['ptr_mate_slct'] == 'w']['ptr_span'])
r_f_span = upper_fence(married[married['ptr_mate_slct'] == 'r']['ptr_span'])

sample_w_m_not = table[(table['mem_sex'] == 'm') & (table['mate_slct'] == 'w')
                        & (table['span'] > w_m_span)]
sample_r_m_not = table[(table['mem_sex'] == 'm') & (table['mate_slct'] == 'r')
                        & (table['span'] > r_m_span)]
sample_w_f_not = table[(table['mem_sex'] == 'f') & (table['mate_slct'] == 'w')
                        & (table['span'] > w_f_span)]
sample_r_f_not = table[(table['mem_sex'] == 'f') & (table['mate_slct'] == 'r')
                        & (table['span'] > r_f_span)]
```

- ✓ 성혼 데이터와 미성혼 데이터를 바탕으로 새로운 변수 생성
- ✓ 생성 변수로 머신러닝 학습

```
# 수치형 변수 생성
data['birth_year_minus'] = - (data['mem_birth_year'] - data['ptr_mem_birth_year'])
data['ann_salary_minus'] = data['mate_ann_salary_p'] - data['ptr_mate_ann_salary_p']
data['property_minus'] = data['possess_property_p'] - data['ptr_possess_property_p']
data['height_minus'] = data['mate_height'] - data['ptr_mate_height']
data['career_minus'] = data['mate_career_p'] - data['ptr_mate_career_p']

# 명목형 변수 생성
data = pd.get_dummies(data, columns = ['mate_slct'])
data = pd.get_dummies(data, columns = ['ptr_mate_slct'])
data = pd.get_dummies(data, columns = ['mem_loc_p'])
data = pd.get_dummies(data, columns = ['ptr_mem_loc_p'])

# 머신러닝 모델 학습
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split

model = LogisticRegression()
X_train, X_test, y_train, y_test = train_test_split(data_model, data_target)
model.fit(X_train, y_train)
pred = model.predict(X_test)
```

07:서비스 프로세스 및 기술소개

세부 프로세스 4단계 <이미지 합성>



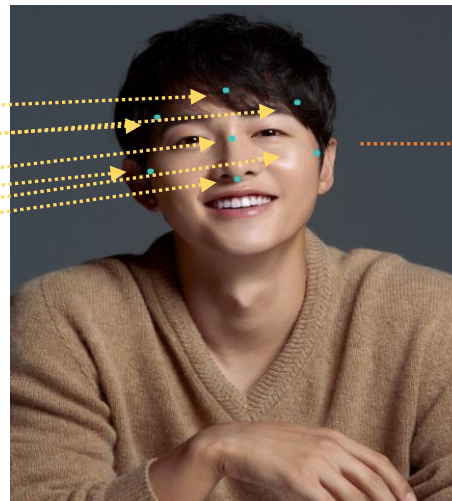
[입력 데이터] TOP3 이성의 이미지 & 동물상 정보 → **이미지 합성 모델** → [결과 데이터] 가면 합성 이미지

MP&CV
이미지
합성

특징점 검출과 원근 변환을 활용하여 이미지 합성

※ 얼굴 각도에 맞춰 가면이 입체적으로 변환

1. (가면) 좌표 추출	2. (이미지) 좌표 추출	3. 가면 이미지 생성	4. 합성 이미지 생성
Mediapipe 좌표 7개 추출	동일 좌표 7개 추출	특징점 검출 & 원근변환 적용	이미지 오버레이 적용



[참고] 관련 코드(요약)

- ✓ Mediapipe Facemesh landmark를 활용하여 가면·사용자 이미지 좌표 추출
- ✓ 해당 좌표를 `Opencv.findhomography`(특징점 추출)과 `warpPerspective`(원근 변환)을 적용하여 사용자 얼굴에 맞게 가면 이미지를 변환

```
import cv2
import mediapipe as mp
import numpy as np
import pandas as pd

# 가면 및 이미지 좌표 추출
ids = [10, 6, 54, 284, 234, 454, 2]

mask_landmark = []

for facial_landmarks in result.multi_face_landmarks :
    for i in ids :
        pt = facial_landmarks.landmark[i]
        x = int(pt.x * width)
        y = int(pt.y * height)
        mask_landmark.append([i,x,y])

df = pd.DataFrame(mask_landmark, columns=['landmark', 'x', 'y'])
```

```
# 변환 가면 이미지 생성
im_src = cv2.imread('mask_image.png', cv2.IMREAD_UNCHANGED)
pts_src = np.array(coordinates, dtype=float)
pts_dst = np.array(landmarks_coordinates, dtype=float)

h, status = cv2.findHomography(pts_src, pts_dst)
im_out = cv2.warpPerspective(im_src, h, ((image.shape[1]),(image.shape[0])))
```

- ✓ 변환된 가면 이미지와 원본 사용자 이미지를 합성하여 새로운 이미지 생성

```
# 이미지 오버레이 적용
def overlay_transparent(background, overlay, x, y):

    background_width = background.shape[1]
    background_height = background.shape[0]

    if x >= background_width or y >= background_height:
        return background

    h, w = overlay.shape[0], overlay.shape[1]
    if x + w > background_width:
        w = background_width - x
        overlay = overlay[:, :w]
    if y + h > background_height:
        h = background_height - y
        overlay = overlay[:h]

    if overlay.shape[2] < 4:
        overlay = np.concatenate(
            [overlay,
             np.ones((overlay.shape[0], overlay.shape[1], 1), dtype=overlay.dtype)*255],
            axis = 2, )

    overlay_image = overlay[..., :3]
    mask = overlay[..., 3:] / 255.0
    background[y:y+h, x:x+w] = (1.0-mask)*background[y:y+h, x:x+w] + mask*overlay_image

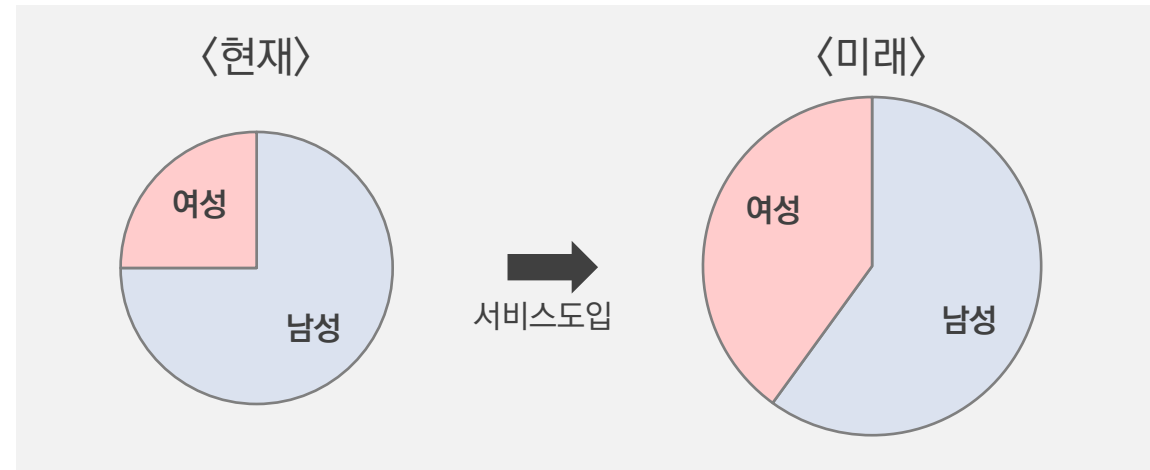
    return background

overlay = overlay_transparent(image, im_out, 0, 0)
```

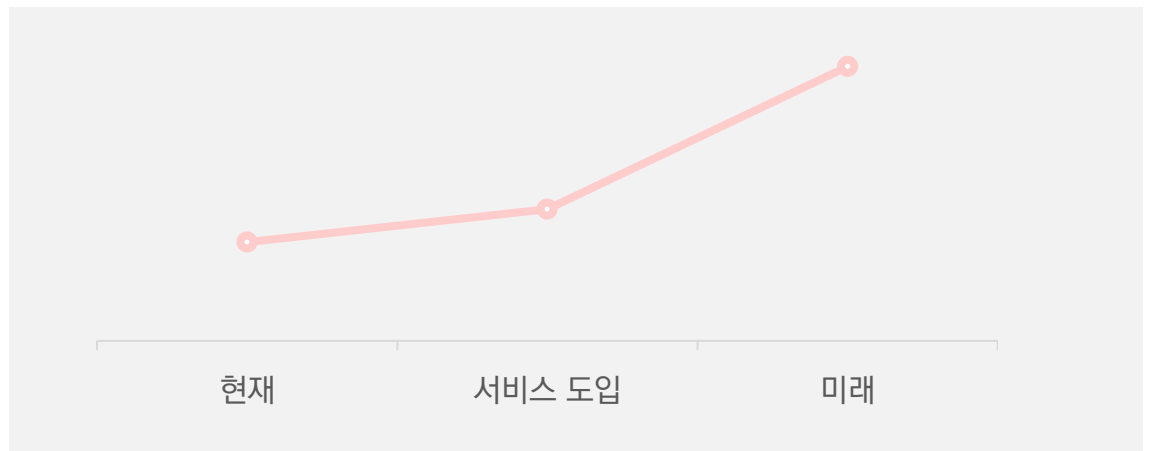
08:기대효과



- 1. 여성 유저 유입 및 활동 증가 → 남녀 비율 불균형 해소



- 2. 남녀 비율 불균형 해소 → 전반적인 서비스 이용률 증가



- 3. 전반적인 서비스 이용률 증가 → 서비스를 통한 성혼율 증가

프로젝트 자체 평가

- 박재운 ● 팀원들과의 협업을 통해 커뮤니케이션 능력을 키울 수 있는 좋은 기회가 되었고, 더 나아가 팀장으로서 리더십을 키울 수 있는 좋은 경험이었습니다.
- 황선진 ● 데이터를 기반으로 아이디어 기획을 하고, 기획에 따라 AI를 적용한 여러 결과물을 만든 멋진 경험이었습니다.
- 강다빈 ● 학습용 데이터가 아닌 실제 회원 데이터를 사용해보니 고려할 사항이 많아 힘들었지만 뿌듯한 경험이었습니다.
- 전해민 ● 평소 가공된 데이터만 보다가, 실무 데이터는 데이터 정제부터 정말 많은 과정이 필요하다는 걸 느끼고 뜻 깊은 경험이었습니다.

참고자료

임일, 「Python을 이용한 개인화 추천 시스템」 청람출판사, 2020

깃허브, “가면 이미지 합성”, https://github.com/kranokdev/Superhero_Mask_Overlay<https://darkpgmr.tistory.com/79>

블로그, “소규모 데이터셋에서 CNN 훈련하기(Kaggle-Dogs vs. Cats)”, <https://foxtrotin.tistory.com/473>

깃허브, “마스크 데이터셋 분류”, <https://github.com/gachonyws/face-mask-detector.git>

취업포털 커리어, “첫 만남시 여성의 외모를 볼 때 어디를 가장 먼저 보는가”, 대상: 직장인 미혼남녀 410명

깃허브 주소 : https://github.com/uiet312/gjs_project

09:추후 프로젝트 제안서

“귀여운 동물 이미지를 활용하여 20대를 타겟으로 한
서로를 알아가는 시뮬레이션 게임 서비스”



여보야가 추천하는 나의 배우자 궁금하신가요?

여보야 추천 시스템이 당신의 배우자를
추천해드립니다!
현재 12,125명이 추천받았습니다.

클릭

