

너의 얼굴은.

딥러닝을 활용한, 안면인식 기반 예측

얼굴의 경로
강찬석
권우진
이경로
정서영

| 목차

- 1 주제 선정
- 2 EDA
- 3 모델 선정 및 시연
- 4 REVIEW
- 5 Q&A

| 주제 선정



| 데이터 가져오기

```
data = os.listdir('./UTKFace/')
df = pd.DataFrame(data=data, columns=['file_name'])
df
```

	file_name
0	100_0_0_20170112213500903.jpg.chip.jpg
1	100_0_0_20170112215240346.jpg.chip.jpg
2	100_1_0_20170110183726390.jpg.chip.jpg
3	100_1_0_20170112213001988.jpg.chip.jpg
4	100_1_0_20170112213303693.jpg.chip.jpg
...	...
23703	9_1_3_20161220222856346.jpg.chip.jpg
23704	9_1_3_20170104222949455.jpg.chip.jpg
23705	9_1_4_20170103200637399.jpg.chip.jpg
23706	9_1_4_20170103200814791.jpg.chip.jpg
23707	9_1_4_20170103213057382.jpg.chip.jpg

kaggle의 UTKFace dataset

| EDA

- 100_ 나이
- _0_ 성별
- _0_ 인종

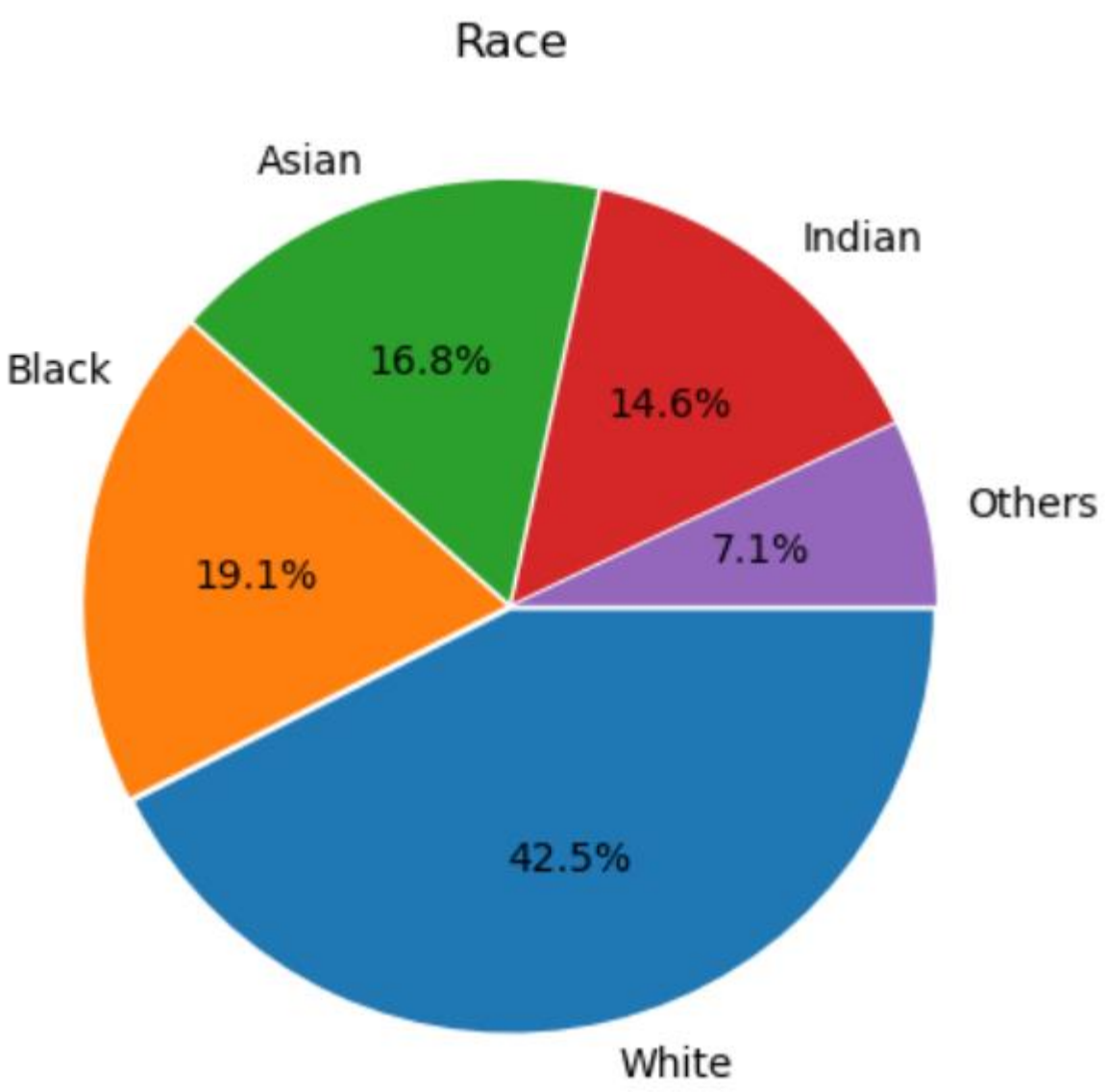
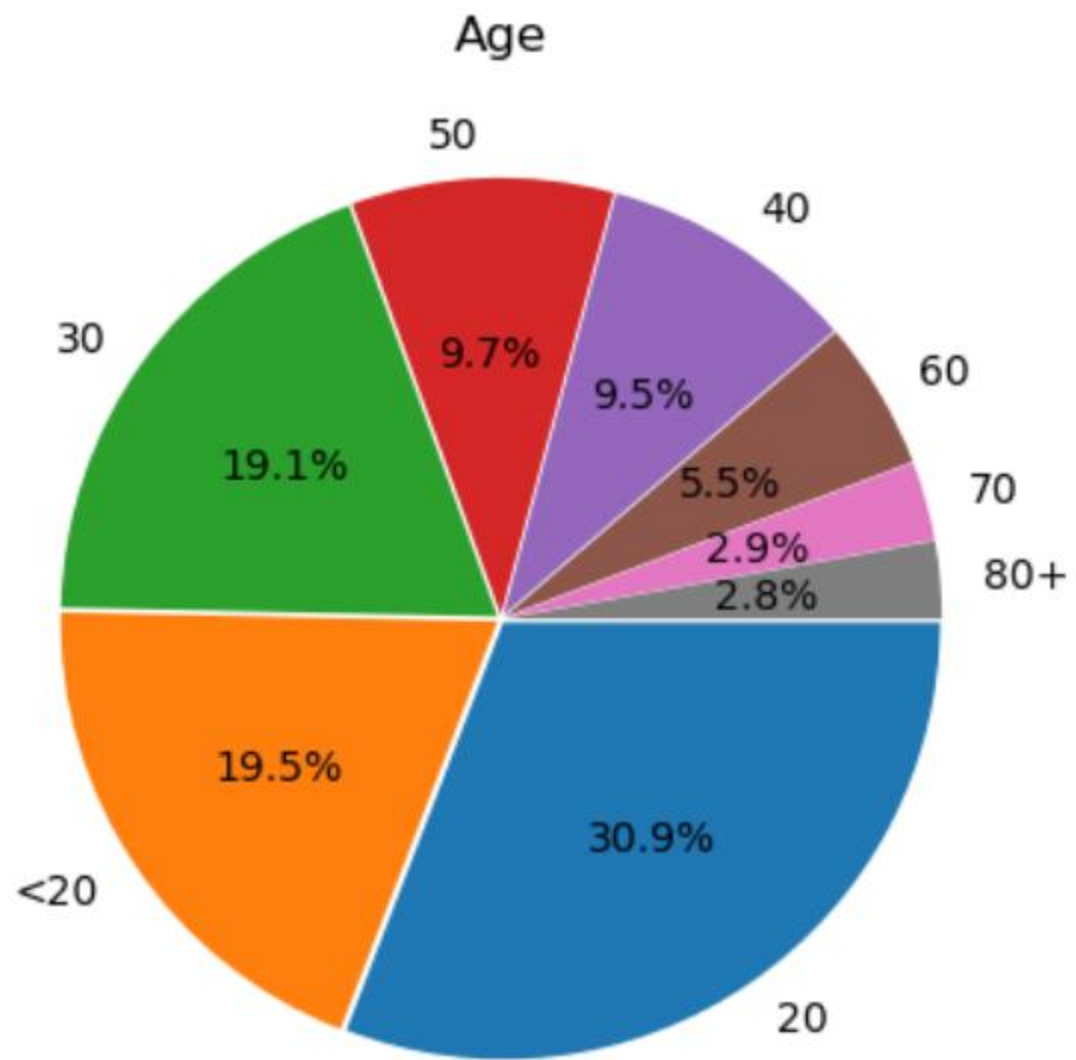
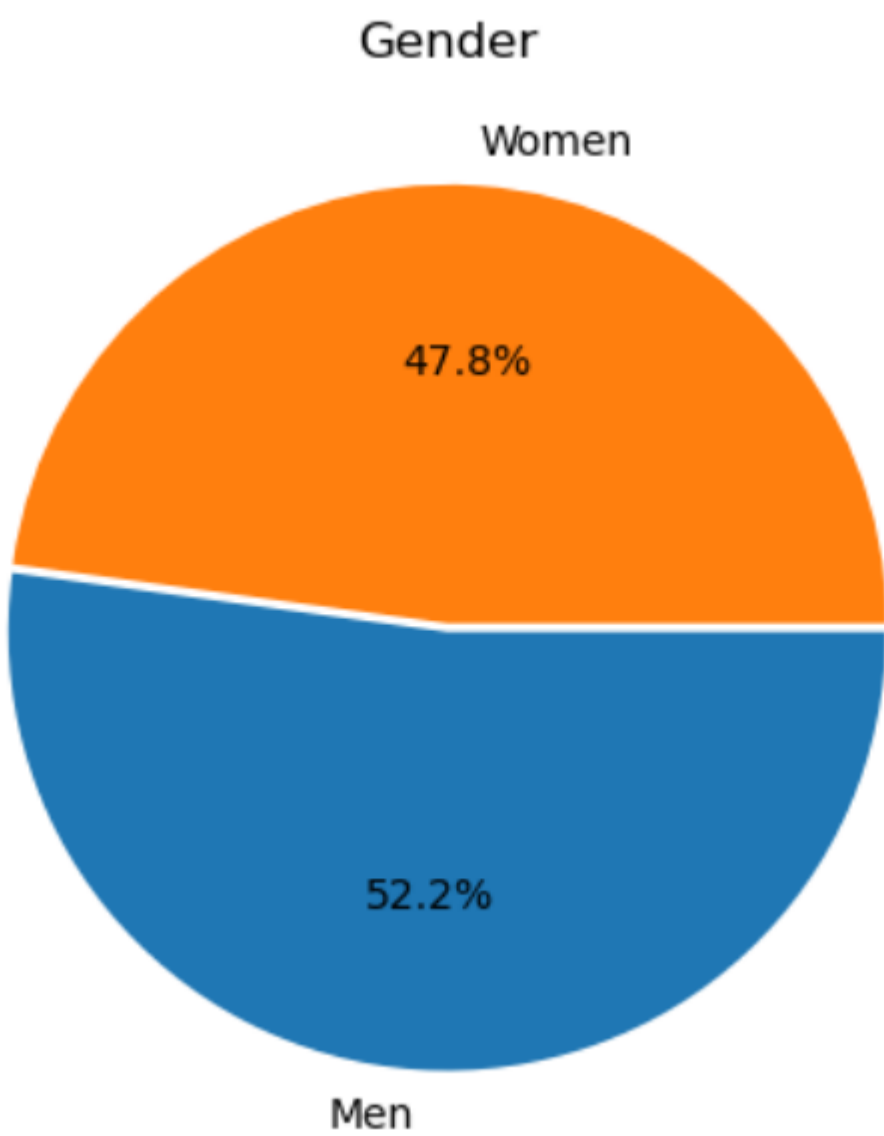
```
data = os.listdir('./UTKFace/')
df = pd.DataFrame(data=data, columns=['file_name'])
df
```

	file_name
0	100_0_0_20170112213500903.jpg.chip.jpg
1	100_0_0_20170112215240346.jpg.chip.jpg
2	100_1_0_20170110183726390.jpg.chip.jpg
3	100_1_0_20170112213001988.jpg.chip.jpg
4	100_1_0_20170112213303693.jpg.chip.jpg
...	...
23514	9_1_3_20161220222856346.jpg.chip.jpg
23515	9_1_3_20170104222949455.jpg.chip.jpg
23516	9_1_4_20170103200637399.jpg.chip.jpg
23517	9_1_4_20170103200814791.jpg.chip.jpg
23518	9_1_4_20170103213057382.jpg.chip.jpg

23707장 → 23518장

EDA

Raw Data



EDA

Race

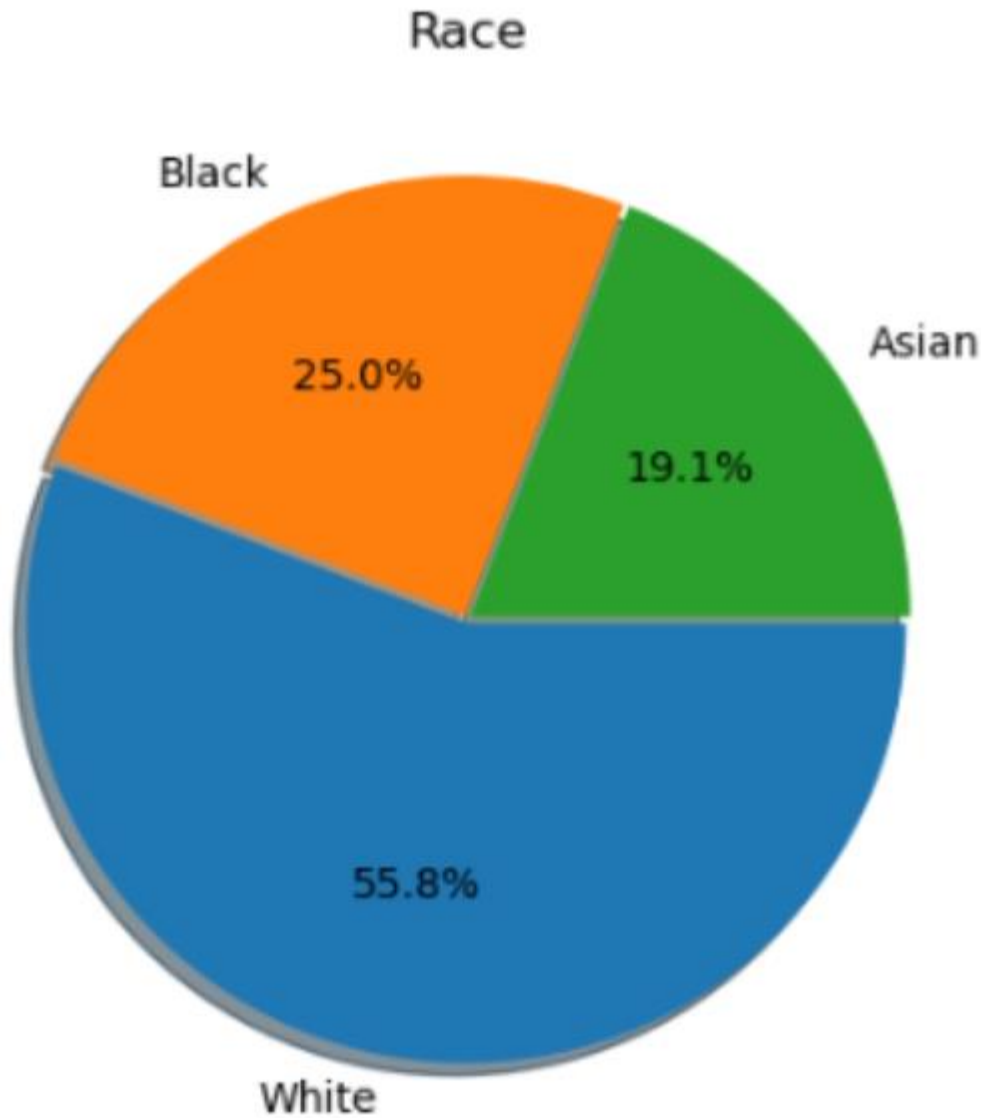
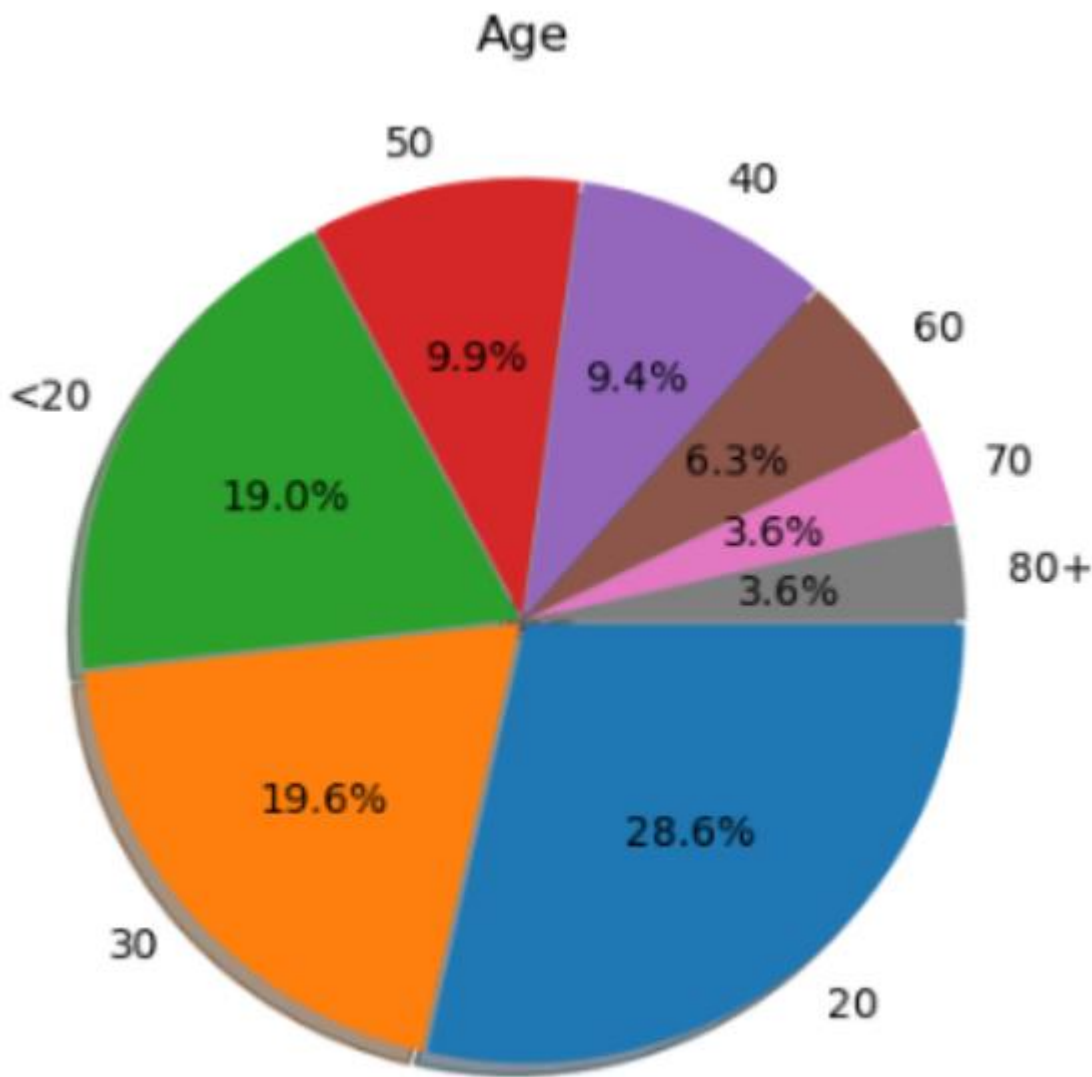
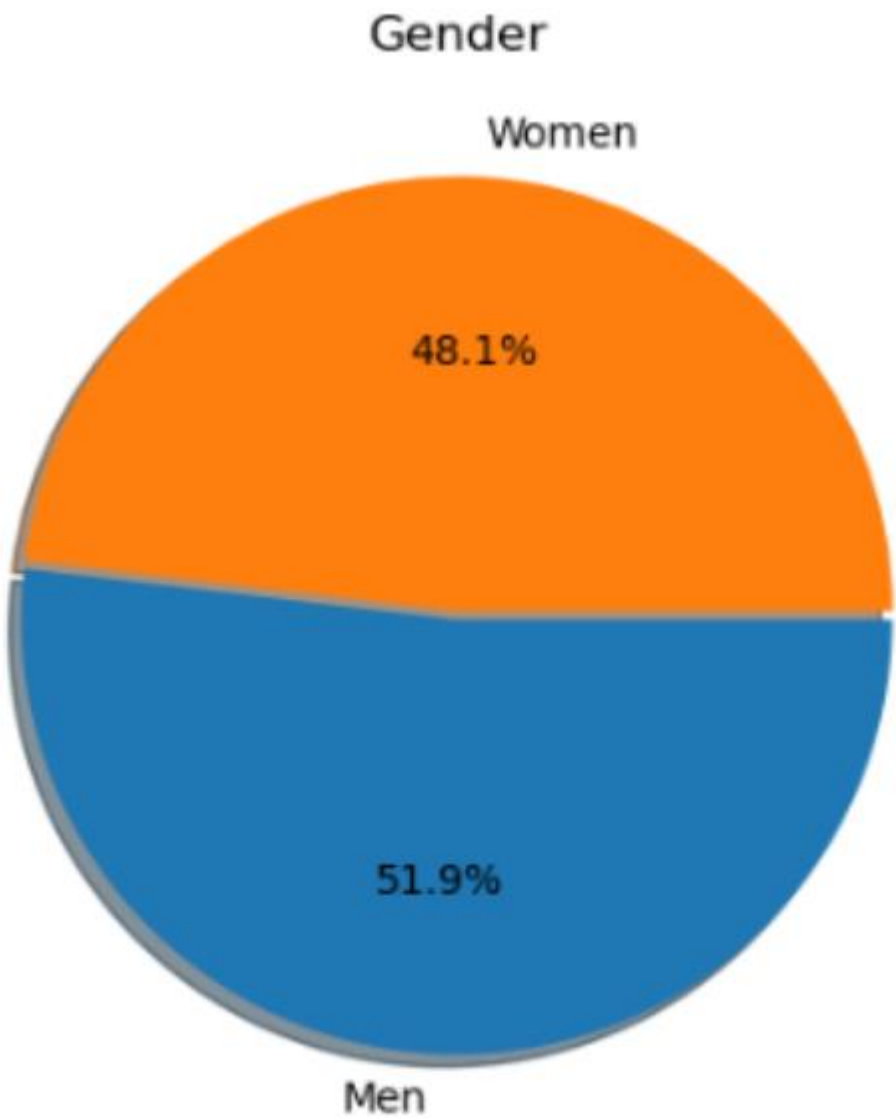
	file_name	age	gender	race
0	100_0_0_20170112213500903.jpg.chip.jpg	100	0	0
1	100_0_0_20170112215240346.jpg.chip.jpg	100	0	0
2	100_1_0_20170110183726390.jpg.chip.jpg	100	1	0
3	100_1_0_20170112213001988.jpg.chip.jpg	100	1	0
4	100_1_0_20170112213303693.jpg.chip.jpg	100	1	0
...
17902	9_1_2_20161219190524395.jpg.chip.jpg	9	1	2
17903	9_1_2_20161219192342173.jpg.chip.jpg	9	1	2
17904	9_1_2_20161219204347420.jpg.chip.jpg	9	1	2
17905	9_1_2_20170102235115156.jpg.chip.jpg	9	1	2
17906	9_1_2_20170104020210475.jpg.chip.jpg	9	1	2
17907 rows	x 4 columns			

	file_name	age	gender	race	age_cat
0	56_1_0_20170117172544019.jpg.chip.jpg	56	1	0	50
1	50_0_0_20170104212134308.jpg.chip.jpg	50	0	0	50
2	45_0_0_20170111204133863.jpg.chip.jpg	45	0	0	40
3	27_0_0_20170114034255912.jpg.chip.jpg	27	0	0	20
4	26_0_0_20170105183712607.jpg.chip.jpg	26	0	0	20
...
10270	9_1_2_20161219190524395.jpg.chip.jpg	9	1	2	<20
10271	9_1_2_20161219192342173.jpg.chip.jpg	9	1	2	<20
10272	9_1_2_20161219204347420.jpg.chip.jpg	9	1	2	<20
10273	9_1_2_20170102235115156.jpg.chip.jpg	9	1	2	<20
10274	9_1_2_20170104020210475.jpg.chip.jpg	9	1	2	<20
10275 rows	x 5 columns				

17907장 → 10275장

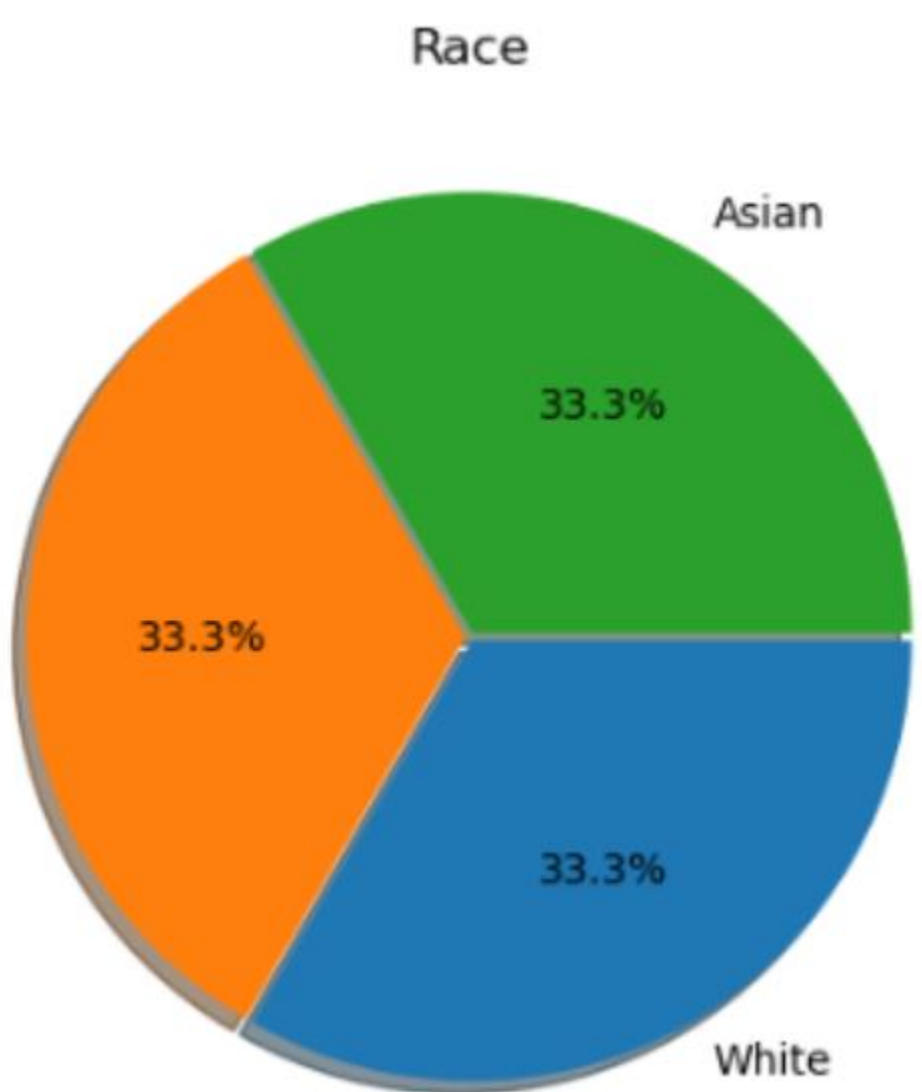
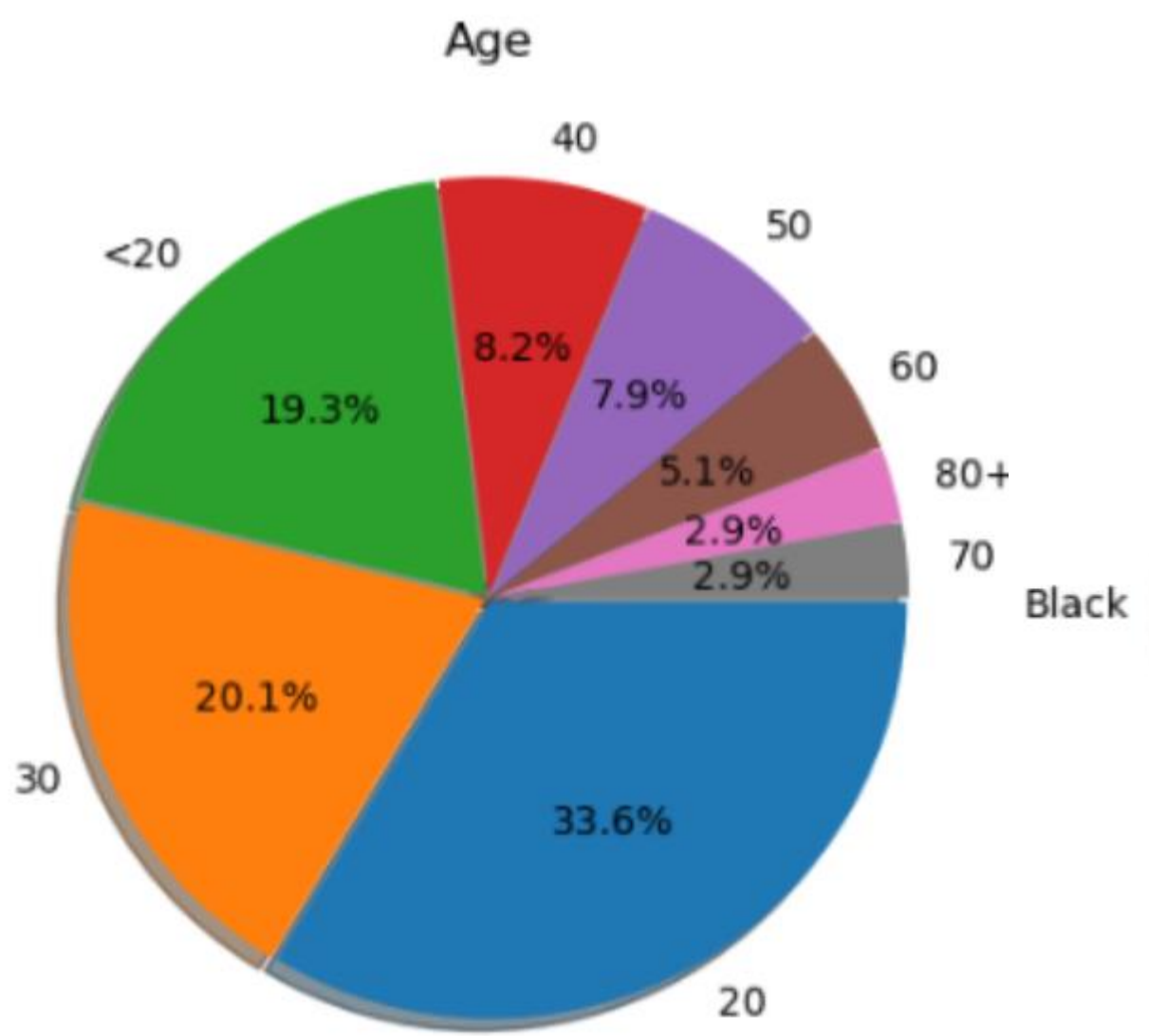
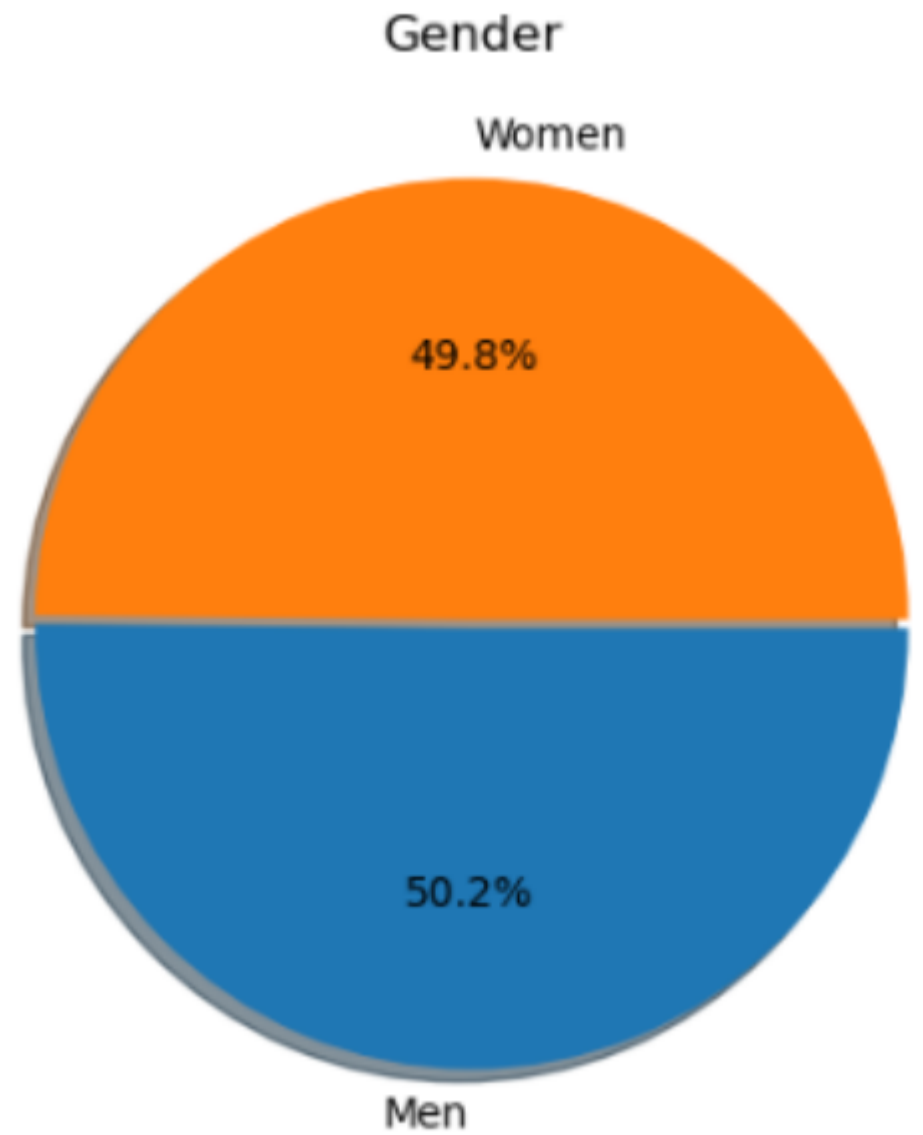
EDA

Undersampling 전



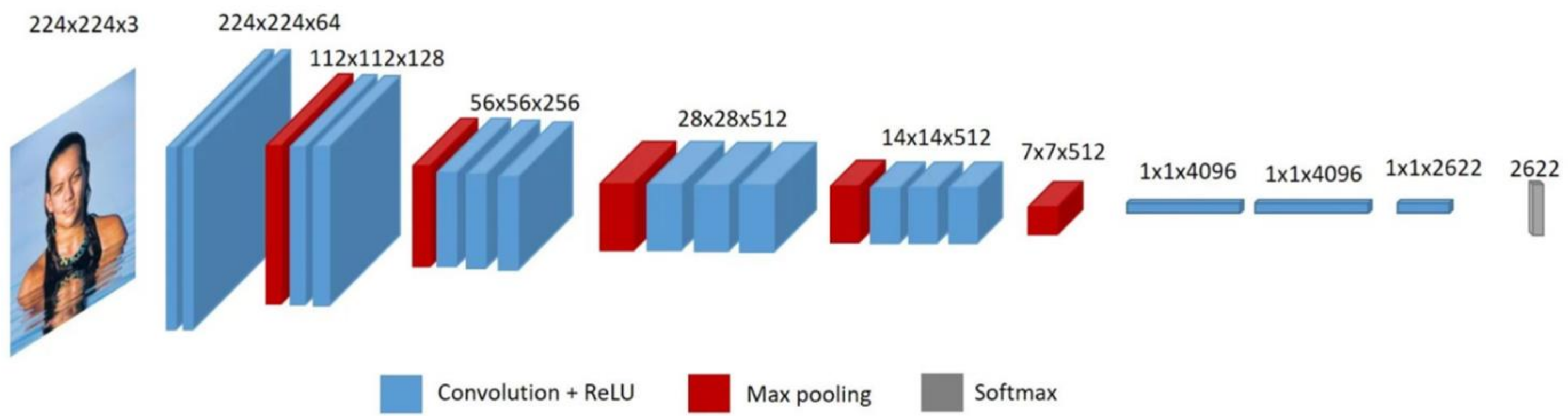
EDA

Undersampling 후



| 모델 선정

VGG Face



| 모델 선정

VGG Face

```
[ ] model_age.summary()
```

Model: "model_2"		
Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
conv1_1 (Conv2D)	(None, 224, 224, 64)	1792
conv1_2 (Conv2D)	(None, 224, 224, 64)	36928
pool1 (MaxPooling2D)	(None, 112, 112, 64)	0
conv2_1 (Conv2D)	(None, 112, 112, 128)	73856
conv2_2 (Conv2D)	(None, 112, 112, 128)	147584
pool2 (MaxPooling2D)	(None, 56, 56, 128)	0
conv3_1 (Conv2D)	(None, 56, 56, 256)	295168
conv3_2 (Conv2D)	(None, 56, 56, 256)	590080
conv3_3 (Conv2D)	(None, 56, 56, 256)	590080
pool3 (MaxPooling2D)	(None, 28, 28, 256)	0

conv4_1 (Conv2D)	(None, 28, 28, 512)	1180160
conv4_2 (Conv2D)	(None, 28, 28, 512)	2359808
conv4_3 (Conv2D)	(None, 28, 28, 512)	2359808
pool4 (MaxPooling2D)	(None, 14, 14, 512)	0
conv5_1 (Conv2D)	(None, 14, 14, 512)	2359808
conv5_2 (Conv2D)	(None, 14, 14, 512)	2359808
conv5_3 (Conv2D)	(None, 14, 14, 512)	2359808
pool5 (MaxPooling2D)	(None, 7, 7, 512)	0
Flatten (Flatten)	(None, 25088)	0
Dropout-1 (Dropout)	(None, 25088)	0
Dense-1 (Dense)	(None, 16)	401424
Dense-2 (Dense)	(None, 8)	136
Output (Dense)	(None, 1)	9

| 모델 선정

```
output_layer = vggface.get_layer('pool5').output
x = Flatten(name='Flatten')(output_layer)
x = Dense(units=512, activation='relu', name='Dense-1')(x)

x = Dropout(rate=0.5, name='Dropout')(x)
x = Dense(units=256, activation='relu', name='Dense-2')(x)

x = Dense(units=1, activation='sigmoid', name='Output')(x)

model_gender = Model(inputs=vggface.input, outputs=x)
```

gender는 이진 분류 ▶ 'sigmoid' 사용

```
output_layer = vggface.get_layer('pool5').output
x = Flatten(name='Flatten')(output_layer)
x = Dense(units=512, activation='relu', name='Dense-1')(x)

x = Dropout(rate=0.5, name='Dropout')(x)
x = Dense(units=256, activation='relu', name='Dense-2')(x)

x = Dense(units=3, activation='softmax', name='Output')(x)

model_race = Model(inputs=vggface.input, outputs=x)
```

race는 다중 분류 ▶ 'softmax' 사용

| 모델 선정

```
output_layer = vggface.get_layer('pool5').output
x = Flatten(name='Flatten')(output_layer)
x = Dropout(rate=0.7, name='Dropout-1')(x)
x = Dense(units=16, activation='relu', name='Dense-1')(x)

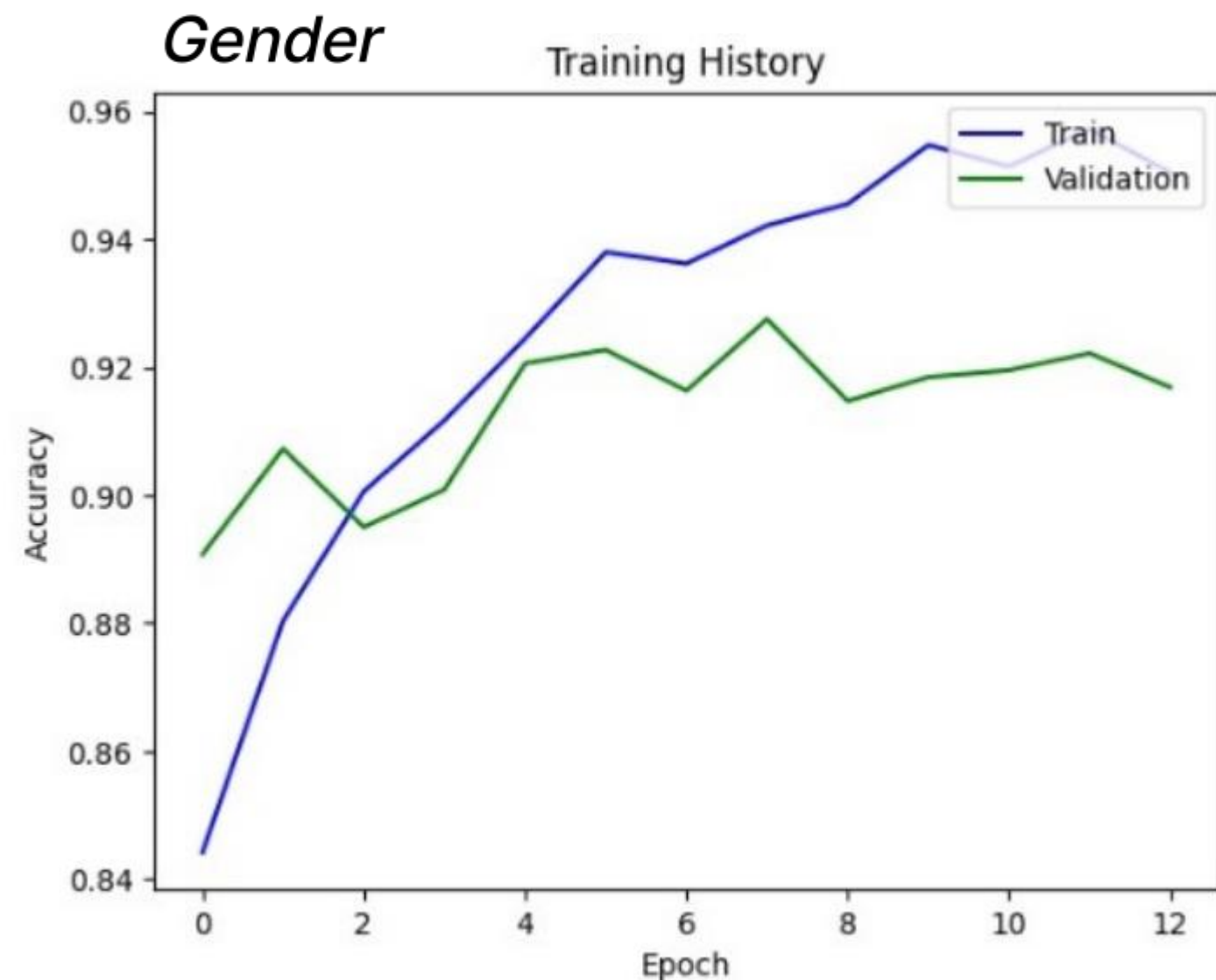
x = Dense(units=8, activation='relu', name='Dense-2')(x)

x = Dense(units=1, activation='linear', name='Output')(x)

model_age = Model(inputs=vggface.input, outputs=x)
```

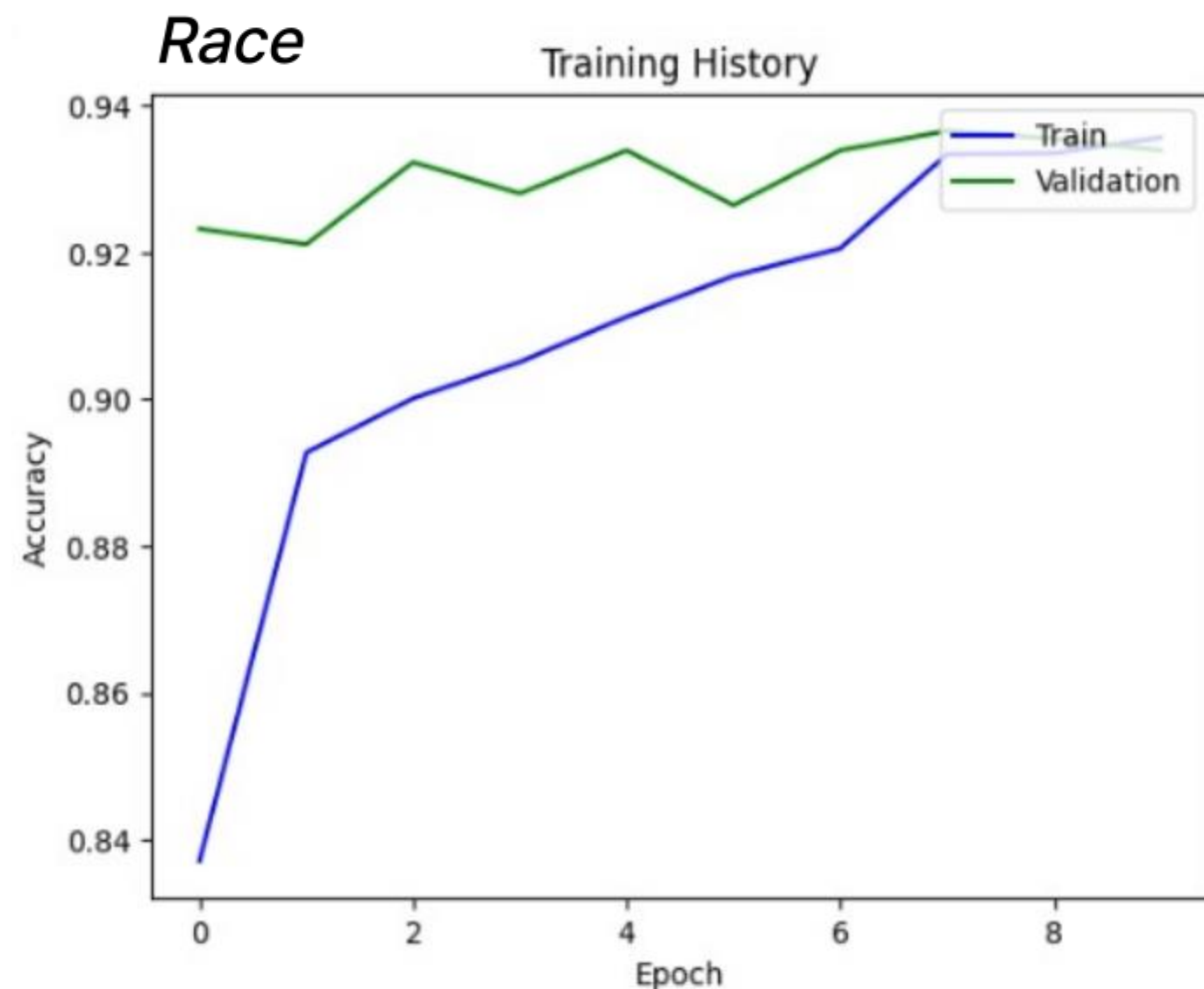
age는 회귀 ► 'linear' 사용

| 성능 평가



```
# 최종적으로 정확도 계산.  
ACC = model_gender.evaluate(test_gender_generator, verbose=0)[1]  
print("Test Accuracy : {}".format(np.round(ACC, 3)))
```

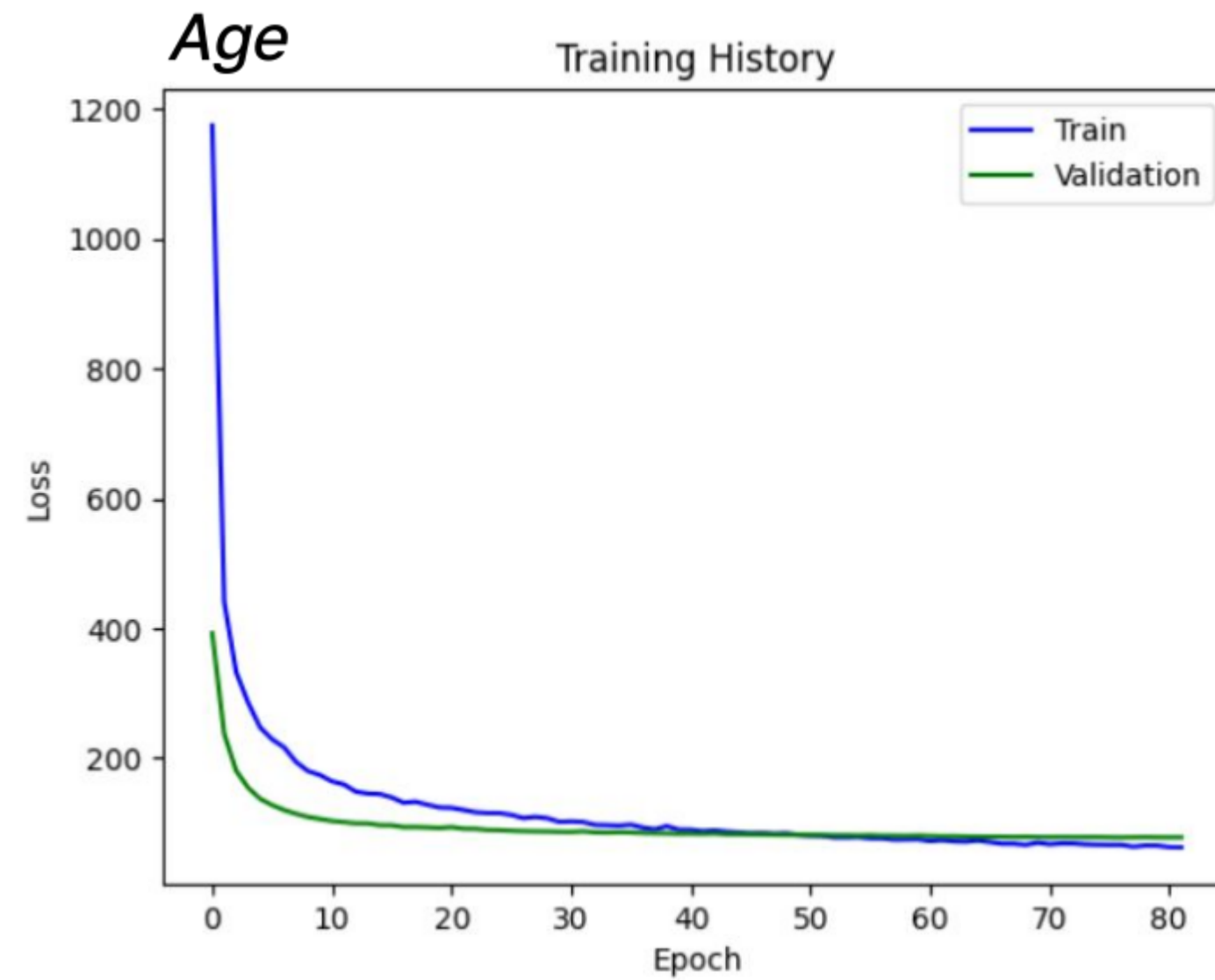
Test Accuracy : 0.917



```
# 최종적으로 정확도 계산.  
ACC = model_race.evaluate(test_race_generator, verbose=0)[1]  
print("Test Accuracy : {}".format(np.round(ACC, 3)))
```

Test Accuracy : 0.934

| 성능 평가



MAE : 6.241
MSE : 72.025
RMSE : 8.487

| REVIEW

age 데이터 불균형

현재 모델로는 단독 사진만 가능

현실적인 제약으로 인한 CNN 모델

한 사람 연령대의 사진들이 아니기에 다소 큰 오차 발생

The left side of the slide features an abstract background composed of several overlapping triangular and quadrilateral shapes in various shades of blue and cyan, creating a dynamic, geometric pattern.

THANK YOU

출처: kaggle UTKFace dataset