

# 헬스퀘어(HealthSquare)

: 사각형 건강남녀가 주목받는 시대

**Team :** 잘살게해주호

Members : 권수현, 김채영, 오주호, 지준희

# 목차

## 1. 프로젝트 개요

- 1-1 프로젝트 선정 배경
- 1-2. 헬스케어(HealthSquare)

## 2. 데이터분석 & 머신러닝

- 2-1. 데이터셋 소개
- 2-2. EDA
- 2-3. 전처리
- 2-4. 머신러닝

## 3. 웹 서비스

- 3-1. 웹 서비스 개발 과정
- 3-2. 웹 서비스 시연

## 4. 프로젝트 결과

- 4-1. 시사점
- 4-2. 개선점

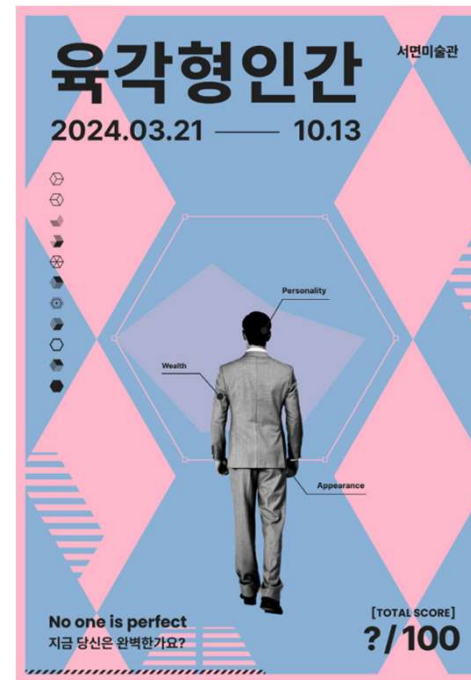
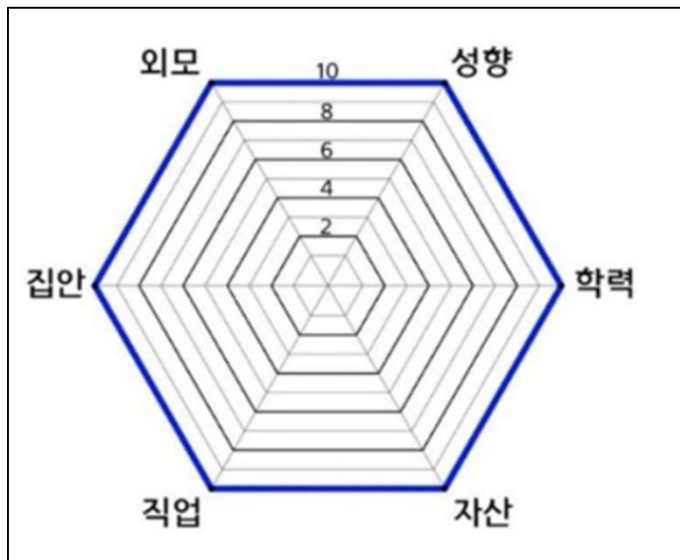
# 1. 프로젝트 개요

1-1 프로젝트 선정 배경

1-2. 헬스케어(HealthSquare)

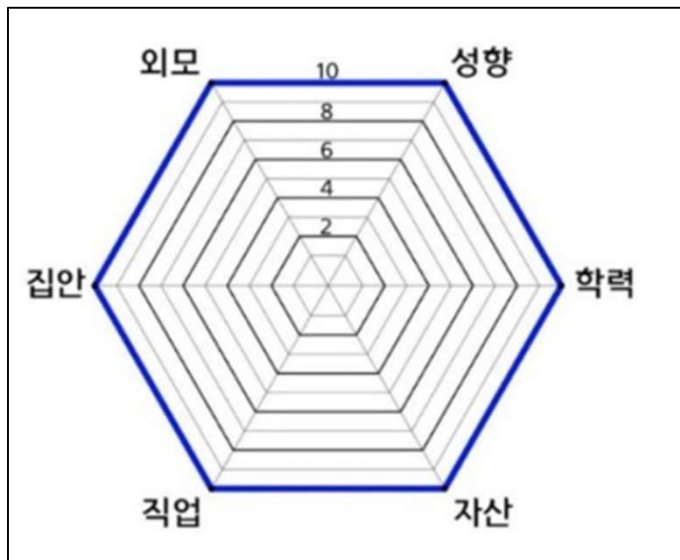
# 1. 프로젝트 개요

육각형 인간?



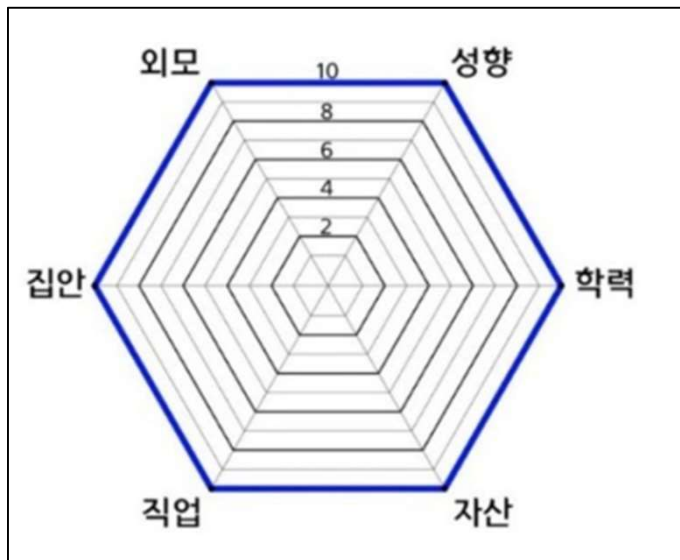
# 1. 프로젝트 개요

육각형 인간 : 모자란 데 없이 전부 평균 이상 하는 인간



# 1. 프로젝트 개요

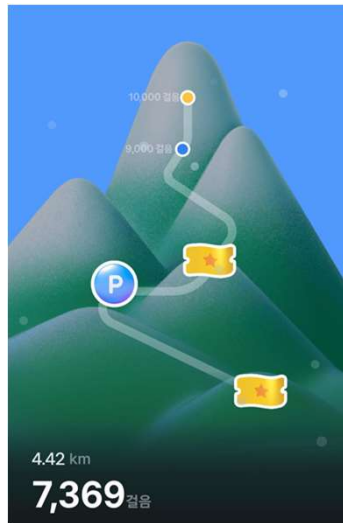
육각형 인간



건강 사각형



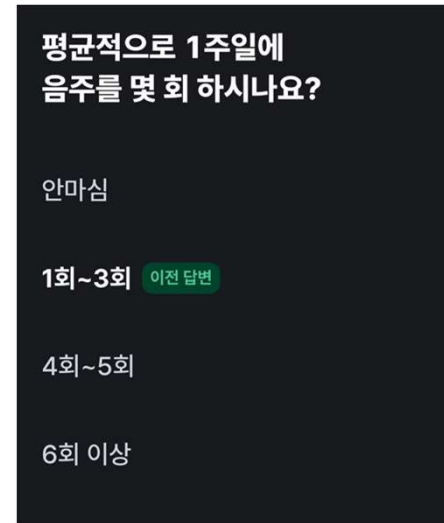
# 1. 프로젝트 개요



토스 만보기  
(운동)



포켓몬슬립  
(수면)



뱅크샐러드  
(음주)

건강기록

기록

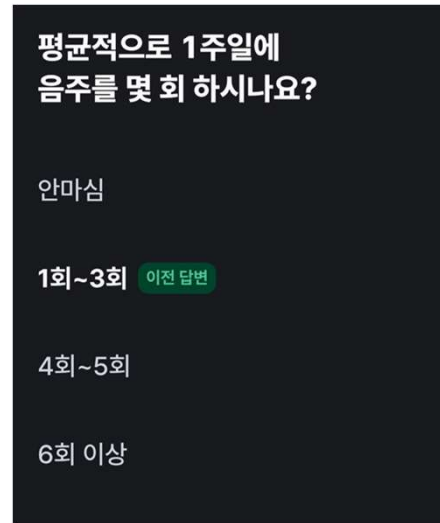
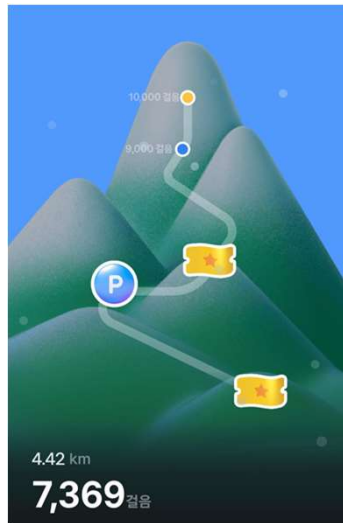
통계

기록하고 포인트 받기

날짜	아침/점심/저녁/간식	BMI
25-03-04 화	+	+
25-03-03 월	495 kcal 오뚜기참깨라면 285 kcal 돌아온 로켓단... 545 kcal 보글보글부대... 0 kcal -	+
25-03-02 일	350 kcal 신라면건면 198 kcal 열알밥한우소... 309 kcal 팬텀의 더블초... 0 kcal -	+
25-03-01 토	339 kcal 망나뇽의 고소... 533 kcal 규동 675 kcal 새우 감바스 0 kcal -	+

손목닥터  
(식사)

# 1. 프로젝트 개요



건강기록

기록

통계

기록하고 포인트 받기

날짜	아침/점심/저녁/간식	BMI
25-03-04 화	+	+
25-03-03 월	495 kcal 오뚜기참깨라면 285 kcal 돌아온 로켓단... 545 kcal 보글보글부대... 0 kcal -	+
25-03-02 일	350 kcal 신라면건면 198 kcal 열알밥한우소... 309 kcal 팬텀의 더블초... 0 kcal -	+
25-03-01 토	339 kcal 망나뇽의 고소... 533 kcal 규동 675 kcal 새우 감바스 0 kcal -	+

건강 데이터를 한번에 관리할 수 있는 서비스는 없을까?  
건강 관리를 꾸준하고 즐겁게 할 수 있는 서비스는 없을까?



## 1-1. 프로젝트 선정 배경

- 1) 건강에 대한 관심 증가
- 2) 개인 맞춤 통합 건강 서비스 부재
- 3) 지속적인 건강 관리를 위한 유인 장치 부족



## 1-2. 헬스케어(HealthSquare)

HealthSquare - AI 헬스케어 어시스턴트

아래 데이터를 입력하고 'Submit' 버튼을 눌러 건강 용량과 추천 리포트를 확인하세요! 이메일 주소를 입력하시면 리포트를 이메일로 받아보실 수 있습니다.

First Name (이름, 선택 사항)	윤희	Flag
Last Name (성, 선택 사항)	지	
Age	29	
Height (cm)	172	
Weight (kg)	61	
Smoking Status (0: Non-smoker, 1: Smoker)	<input checked="" type="radio"/> 0 <input type="radio"/> 1	
Exercise Frequency (times/week)	7	
Diet Quality (0-100)	75	
Sleep Hours (0-12)	5	
Alcohol Consumption (drinks/week)	1	
현재 앓고 계신 질병 또는 건강상의 문제 (선택 사항)	없음	
리포트 수신 이메일 주소 (선택 사항)	jh2boy@naver.com	
Clear	Submit	

사각형 건강남녀로 만들어주는 케어 서비스, 헬스케어

## 1-2. 헬스퀘어(HealthSquare)

### 활용 방안

#### 1) Check

-> 본인의 건강 등급을 리포트로 제공하여 시각적으로 파악 가능

#### 2) Compare

-> 동일 연령대와 본인의 건강 지표를 비교하여 부족한 영역 보완 가능

#### 3) Coaching

-> 건강 등급을 개선할 수 있는 방법과 상황에 맞는 건강 관련 제품 추천 가능

#### 4) Coupon

-> 건강 목표를 설정하고 달성할 경우 건강 관련 제품 할인 쿠폰 수령 가능 (베타서비스)

## 2. 데이터분석 & 머신러닝

2-1. 데이터셋 소개

2-2. EDA

2-3. 전처리

2-4. 머신러닝

## 2-1. 데이터셋 소개

<https://www.kaggle.com/datasets/pratikyuvrajchougule/health-and-lifestyle-data-for-regression>

검색

### Health and Lifestyle Data for Regression

Dataset for Predicting Health Scores Using Lifestyle Factors



[Data Card](#) [Code \(10\)](#) [Discussion \(0\)](#) [Suggestions \(0\)](#)

#### About Dataset

This dataset is designed to study the impact of various lifestyle factors on an individual's health score using linear regression analysis. It provides a comprehensive view of factors such as age, BMI, exercise frequency, diet quality, sleep hours, smoking status, and alcohol consumption. The dataset can be used for exploratory data analysis, machine learning models, and understanding correlations between health and lifestyle.

#### Columns Description

- ☐ **Age** : 나이
- ☐ **BMI** :  $(\text{몸무게(kg)}) / (\text{키(m)})^2$
- ☐ **Exercise\_Frequency** : 운동횟수(categorical, values 0-7)
- ☐ **Diet\_Quality** : 식사점수(continuous, 0-100)
- ☐ **Sleep\_Hours** : 수면시간
- ☐ **Smoking\_Status** : 흡연여부(0 = Non-smoker, 1 = Smoker)
- ☐ **Alcohol\_Consumption** : 음주량
- ☐ **Health\_Score(Target)** : 건강점수(continuous, 0-100)

#### Usability

8.24

#### License

MIT

#### Expected update frequency

Not specified

#### Tags

Regression

Health and Fitness

#### Dataset statistics

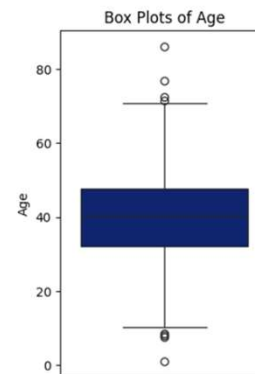
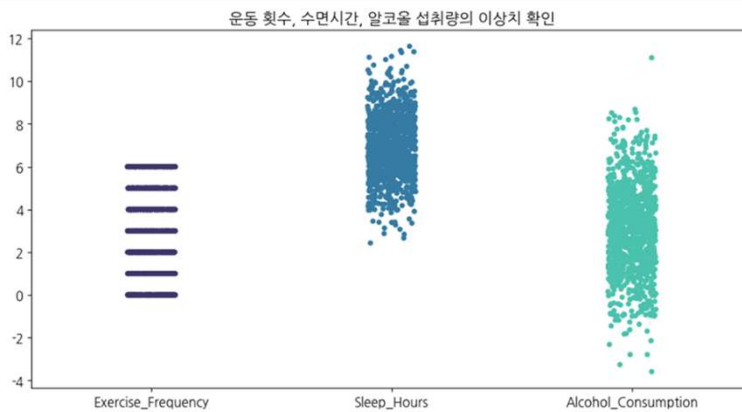
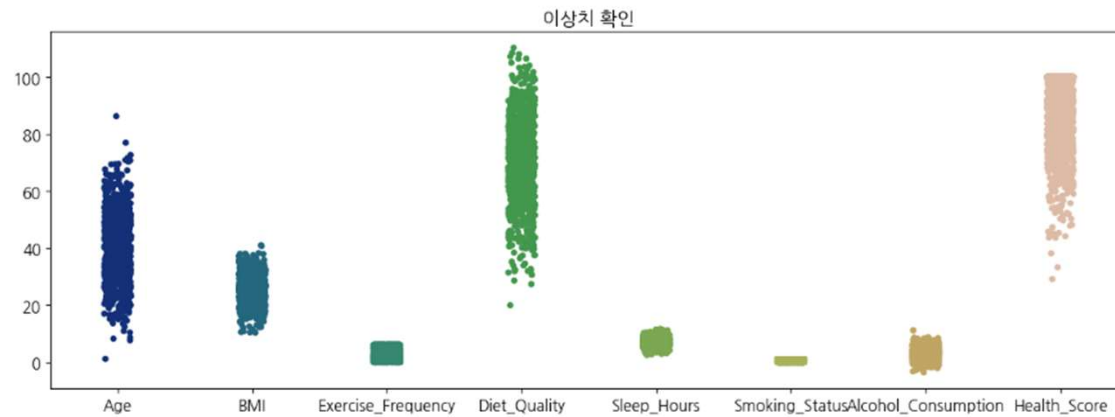
Number of variables	8
Number of observations	1000
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%

#### Variable types

	Numeric	Categorical
	7	1
0	Age	1000 non-null float64
1	BMI	1000 non-null float64
2	Exercise_Frequency	1000 non-null int64
3	Diet_Quality	1000 non-null float64
4	Sleep_Hours	1000 non-null float64
5	Smoking_Status	1000 non-null int64
6	Alcohol_Consumption	1000 non-null float64
7	Health_Score	1000 non-null float64

dtypes: float64 (6), int64 (2)  
memory usage: 62.6 KB

## 2-2. EDA



### 결측치 확인

전체 데이터 1,000개 중 결측치 없음



### 이상치 확인

Age: 신생아 데이터 발견

Diet\_Quality: 만점인 100점 초과값 존재

Alcohol\_Consumption: 음수값 존재



### 데이터 단위 확인

변수간 단위 차이 큼

## 2-3. 전처리 과정1

### 스케일링 + 특성공학을 통해 독립변수 범주화

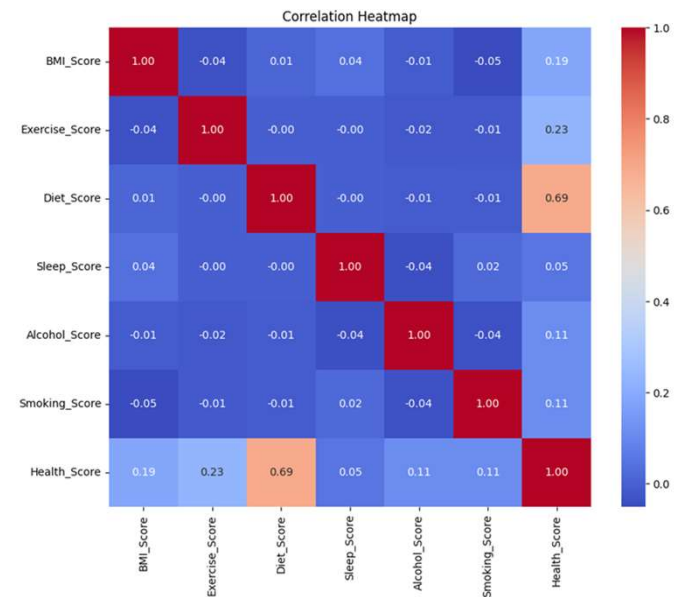
```

1 import numpy as np
2
3 def categorize_bmi(bmi):
4     if 10 <= bmi < 11: return 1
5     elif 11 <= bmi < 12: return 2
6     elif 12 <= bmi < 13: return 3
7     elif 13 <= bmi < 14: return 4
8     elif 14 <= bmi < 15: return 5
9     elif 15 <= bmi < 16: return 6
10    elif 16 <= bmi < 17: return 7
11    elif 17 <= bmi < 18: return 8
12    elif 18 <= bmi < 18.5: return 9
13    elif 18.5 <= bmi < 25: return 10
14    elif 25 <= bmi < 27.5: return 9
15    elif 27.5 <= bmi < 30: return 8
16    elif 30 <= bmi < 32.5: return 7
17    elif 32.5 <= bmi < 35: return 6
18    elif 35 <= bmi < 36: return 5
19    elif 36 <= bmi < 37: return 4
20    elif 37 <= bmi < 38: return 3
21    elif 38 <= bmi < 39: return 2
22    else:
23        return 1
24
25 df['BMI'] = df['BMI'].apply(categorize_bmi)

```

	0	1
Age	46.0	38.0
BMI	7.0	8.0
Exercise_Frequency	10.0	10.0
Diet_Quality	6.0	4.0
Sleep_Hours	10.0	10.0
Smoking_Status	0.0	1.0
Alcohol_Consumption	8.0	3.0
Health_Score	71.0	57.0

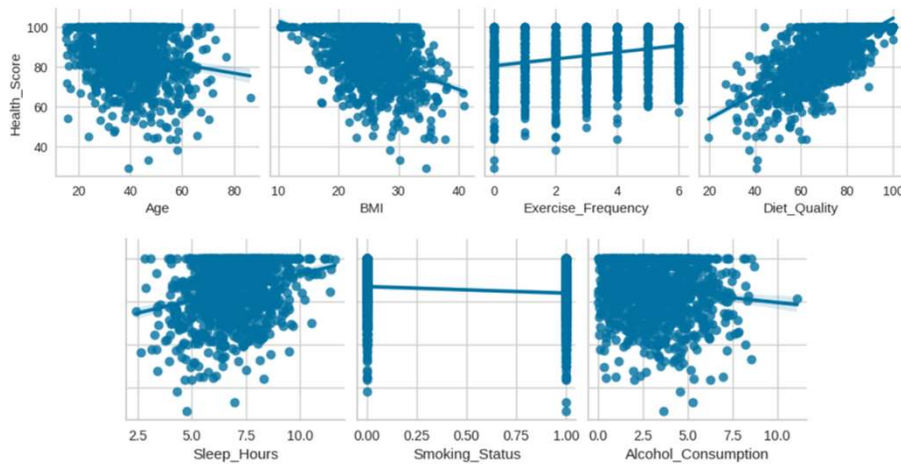
스케일링 개념으로 정상 범위를 10점 만점으로 설정하고 벗어나면 1~2점씩 점수를 감소시켜서 데이터 자체를 범주로 변형함 (나이, 흡연 여부 제외)



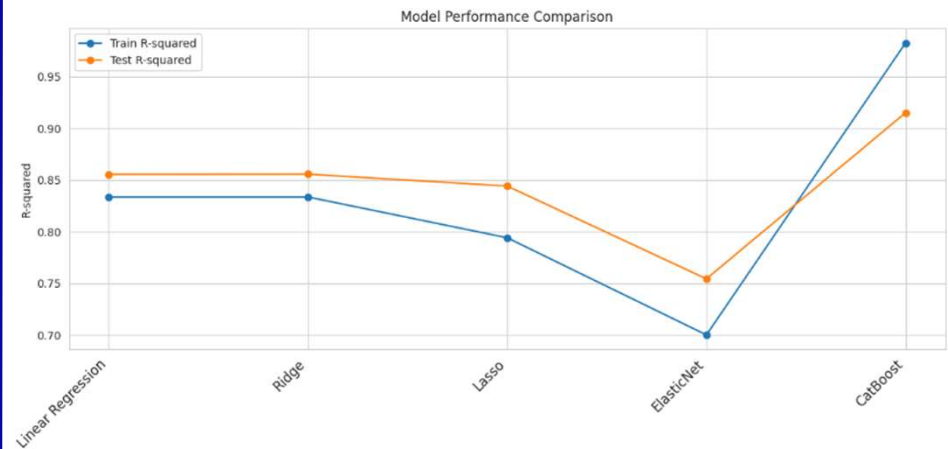
Diet\_Quality 의 상관관계만 높았기 때문에 건강지표 산출이 객관적이지 않음

## 2-3. 전처리 과정2

### 다항회귀 특성 공학 후 규제용 회귀 모델 적용



변수들의 관계가 선형적/비선형적인 것이 혼합되어 있어서  
비선형적 관계를 다루는 다항회귀 적용함

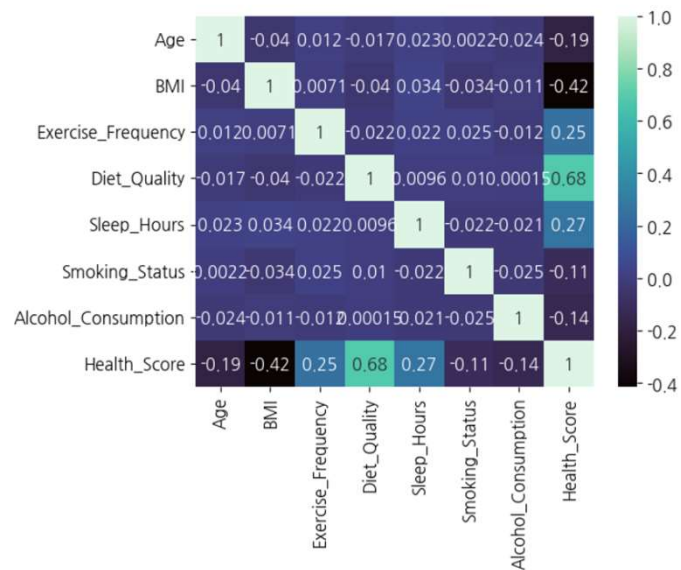


나쁘지 않은 성능을 보이지만, 산점도 그래프를 통해 선형적  
관계가 있는 것처럼 보여서 다른 방법으로도 시도함

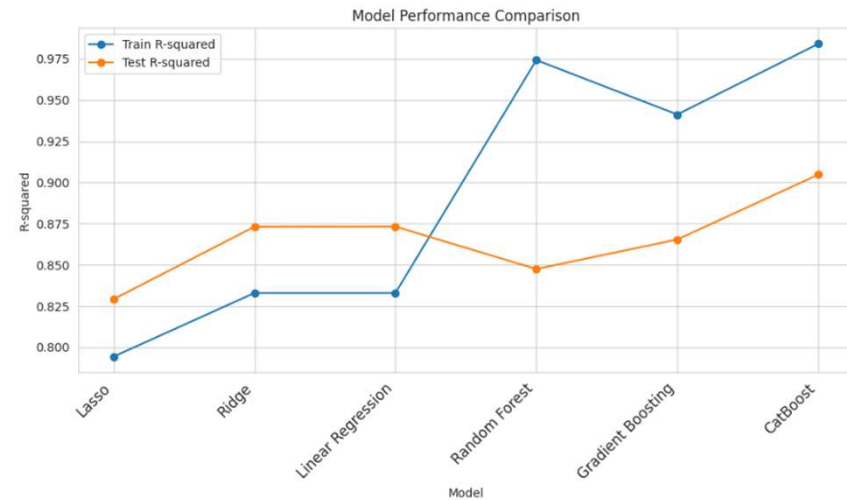


## 2-3. 전처리 과정3

### 표준화 및 파라미터 조정



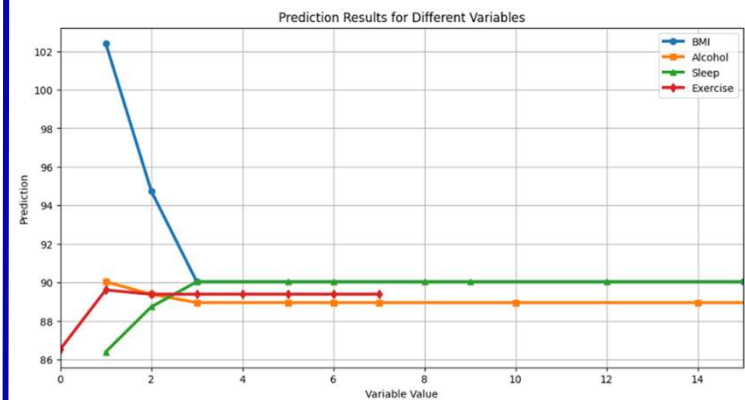
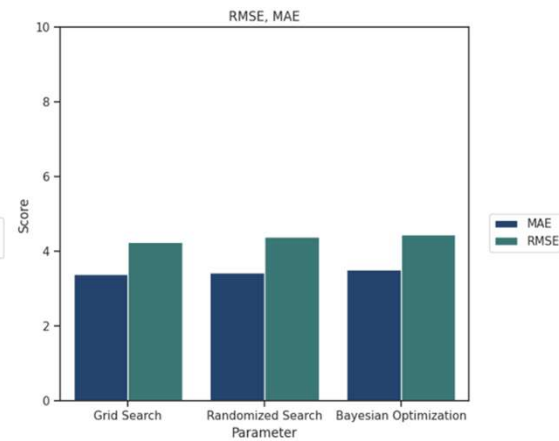
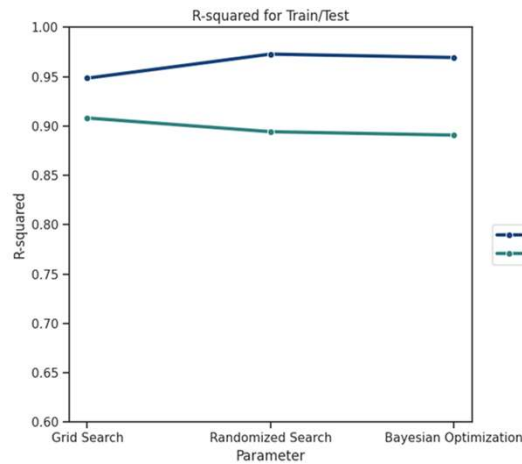
- 변수 간 단위 차이 조정을 위해 표준화 진행
- 스케일링 후 여전히 target 과 상관성이 높음
- 가장 결정계수가 높았던 Cat Boost 을 사용



과대적합 발생

## 2-3. 전처리 과정3

### 표준화 및 파라미터 조정



하이퍼파라미터 설정

(Grid Search CV, Randomized Search, Bayesian Optimization)

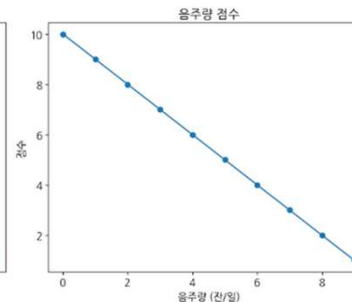
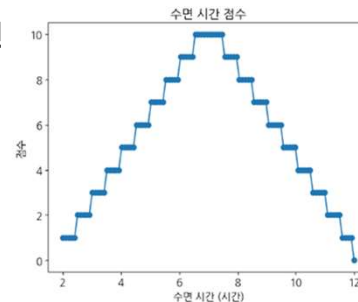
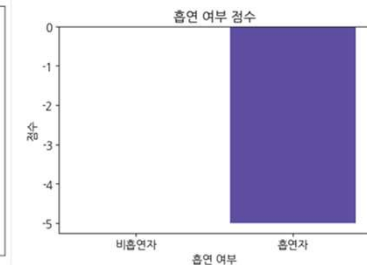
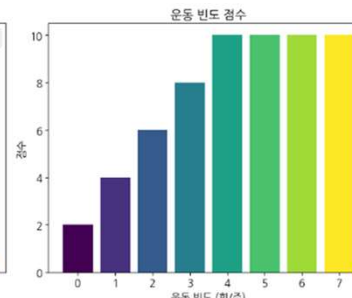
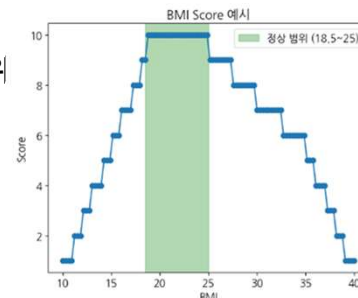
- 사용자 입력 데이터 중 하나라도 값이 2 이상이면 건강지표 점수가 동일한 문제  
=> 스케일링을 적용해 해결
- 타겟값 편중 및 타겟값 자체가 잘못 매겨졌기 때문에 학습시켜도 좋은 결과가 나오지 않는 문제

## 2-4. 최종 모델 선정

### 새로운 타겟 생성 후 분류 모델 적용

- 각 열마다 최고의 건강 상태 유지를 위한 습관의 **적정범위**를 10점 만점으로 설정하고 범위에서 벗어날 때마다 1~2점씩 점수를 감소시킨 **Score 열** 생성함
- 이 열들의 값을 **누적으로** 합해서 새로운 타겟열인 **'New Health Score'**에 저장함
- **New Health Score** 점수를 10단위로 구간으로 나눠서 건강 등급을 매기는 'Health Class'를 생성함
- 회귀 말고 **분류 모델** 구축

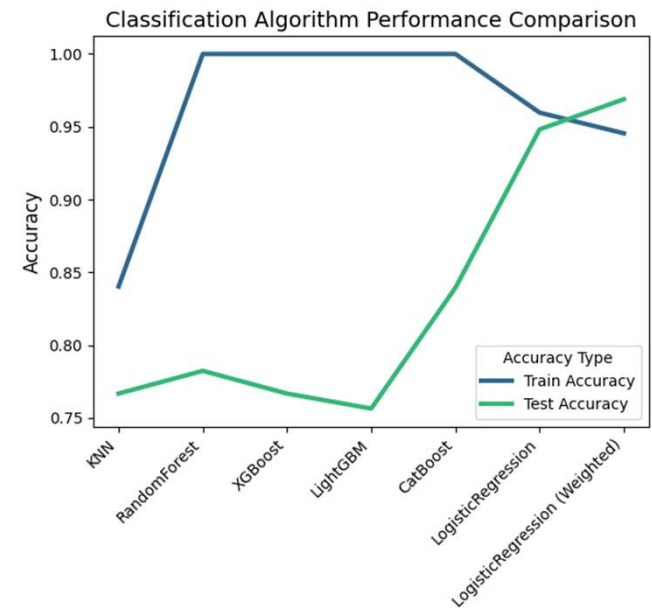
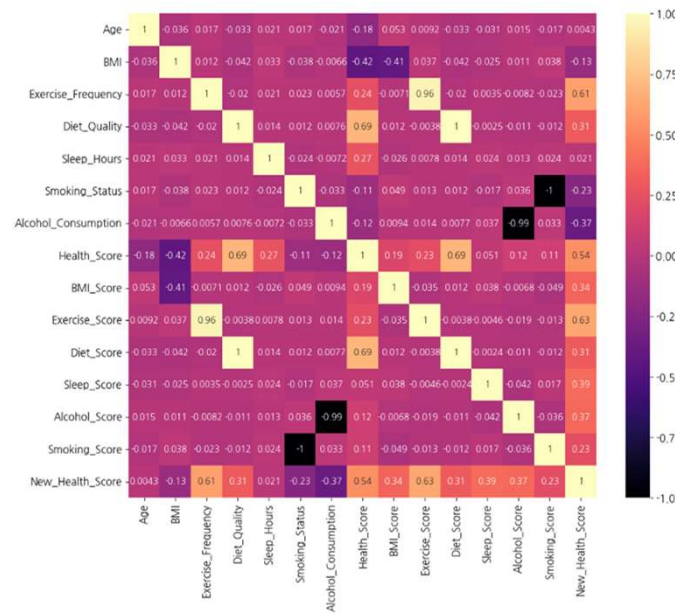
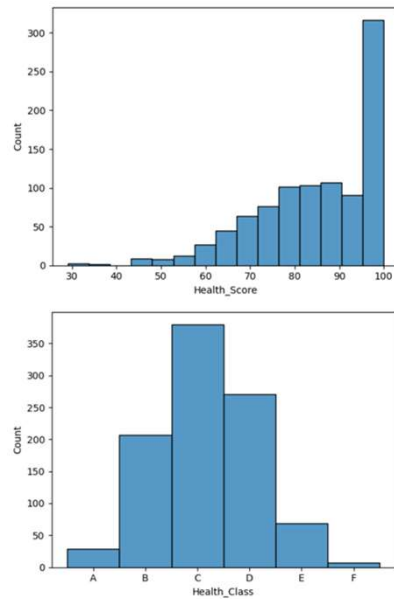
$$\text{New\_Health\_Score} = (\text{BMI\_Score} * 2 + \text{Exercise\_Score} * 2 + \text{Diet\_Score} * 2 + \text{Sleep\_Score} * 2 + \text{Alcohol\_Score} * 2) + \text{Smoking\_Score}$$



변수	정상 범위
BMI	18.5~25
수면 시간	6.5~7.5시간
음주량	0~1번/일
운동 빈도	4번 이상
흡연 여부	비흡연자

## 2-4. 최종 모델 선정

### 새로운 타겟 생성 후 분류 모델 적용



타겟 편중값을 고르게 분포시켜서 여러 점수에 대해서 모델 학습이 잘 되게 함 => 객관적인 건강지표 도출

## 3. 웹 서비스

3-1. 웹 서비스 개발 과정

3-2. 웹 서비스 시연

### 3. 웹 서비스



#### Gradio

Python 코드 몇 줄로 간단한 웹 UI를 만들고, 머신러닝 모델/데이터 분석 기능을 쉽게 배포할 수 있는 라이브러리

#### Gradio의 주요 기능

- **빠른 웹 UI 생성:** 머신러닝 모델, 데이터 분석 함수 등을 몇 줄의 코드로 배포 가능
- **다양한 입력/출력 지원:** 텍스트, 이미지, 오디오, 비디오 등 다양한 데이터 타입을 지원
- **인터랙티브한 테스트 환경:** 사용자 입력을 실시간으로 반영 가능 (`live=True`)
- **로컬 & 공유 가능:** 개발 PC에서 실행하거나, 공유 가능한 링크를 자동 생성해서 배포 가능

## 3-1. 웹 서비스 개발 과정

First Name (이름, 선택 사항)  
예: 길동

Last Name (성, 선택 사항)  
예: 홍

Age  
28

Height (cm)  
184

Weight (kg)  
65

Smoking Status (0: Non-smoker, 1: Smoker)  
☒ 0 ☐ 1

Exercise Frequency (times/week)  
0 4 7

Diet Quality (0-100)  
0 46 100

Sleep Hours (0-12)  
0 7 12

Alcohol Consumption (drinks/week)  
0 5 10

현재 앓고 계시는 질병 또는 건강상의 문제 (선택 사항)  
예: 당뇨병, 다이어트 중

리포트 수신 이메일 주소 (선택 사항)  
예: example@email.com

### 입력 인터페이스

- 1) Name -> 사용자 이름 입력
- 2) Age -> 연령 평균 건강 지표 산출
- 3) Height, Weight -> BMI(키/몸무게<sup>2</sup>) 계산
- 4) Smoking -> 흡연 여부 이진 분류
- 5) Exercise, Diet, Sleep, Alcohol -> 핵심 건강 지표
- 6) 이메일 -> 건강 결과 전송

## 3-1. 웹 서비스 개발 과정

First Name (이름, 선택 사항)

예: 길동

Last Name (성, 선택 사항)

예: 홍

Age

28

Height (cm)

184

Weight (kg)

65

Smoking Status (0: Non-smoker, 1: Smoker)

☒ 0 ☐ 1

Exercise Frequency (times/week)

0 4 7

Diet Quality (0-100)

0 46 100

Sleep Hours (0-12)

0 7 12

Alcohol Consumption (drinks/week)

0 5 10

현재 알고 계시는 질병 또는 건강상의 문제 (선택 사항)

예: 당뇨병, 다이어트 중

리포트 수신 이메일 주소 (선택 사항)

예: example@email.com

```
demo = gr.Interface(
    fn=predict_health_score,
    inputs=[
        gr.Textbox(label="First Name (이름, 선택 사항)", placeholder="예: 길동", value=""),
        gr.Textbox(label="Last Name (성, 선택 사항)", placeholder="예: 홍", value=""),
        gr.Number(label="Age"),
        gr.Number(label="Height (cm)"),
        gr.Number(label="Weight (kg)"),
        gr.Radio([0, 1], label="Smoking Status (0: Non-smoker, 1: Smoker)"),
        gr.Slider(0, 7, step=1, label="Exercise Frequency (times/week)"),
        gr.Slider(0, 100, step=1, label="Diet Quality (0-100)"),
        gr.Slider(0, 12, step=0.5, label="Sleep Hours (0-12)"),
        gr.Slider(0, 10, step=0.5, label="Alcohol Consumption (drinks/week)"),
        gr.Textbox(label="현재 알고 계시는 질병 또는 건강상의 문제 (선택 사항)", placeholder="예: 당뇨병, 다이어트, 근육통, 스트레스 등", value=""),
        gr.Textbox(label="리포트 수신 이메일 주소 (선택 사항)", placeholder="예: example@email.com", value="")
    ],
)
```

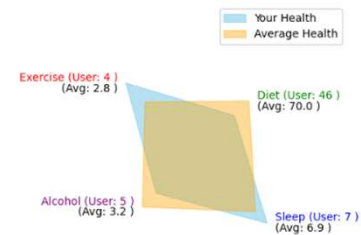


## 3-1. 웹 서비스 개발 과정

### Check - 건강 상태 확인

• 당신의 건강 등급: C

### Compare - 연령대 비교



• 동일 연령대(20.0대)와 비교:

- 운동 빈도: 당신 4회/주 (평균: 2.8회/주, +1.2회) 🏃
- 식단 질: 당신 46 (평균: 70.0, -24.0) 🍌
- 수면 시간: 당신 7시간 (평균: 6.9시간, +0.1시간) 😴
- 음주량: 당신 5회/주 (평균: 3.2회/주, +1.8회) 🍷

• 식단 질이 낮아졌네요. 영양소가 풍부한 식품을 추가해보세요. 예를 들어, 아침에 아보카도 토스트나 오트밀을 드셔보세요. 🥑  
• 음주량이 주당 5회로 높은 편입니다. 음주를 줄이고, 대신 물이나 허브티를 드셔보세요. 예를 들어, 허브 티를 우려는 상쾌한 대안이 될 수 있습니다. 🍵

📌 Coaching - 건강 개선 팁

• 건강 문제 입력이 없으므로 앞으로 건강 사각형을 잘 유지하십시오. 🏆

📌 Coupon - 목표 설정 및 꾸준한 제공 (베타 서비스 준비 중)

- 목표를 설정하고 달성하면 쿠폰을 드립니다. 현재 베타 서비스 준비 중입니다. 🎫
- 예시: 1달 동안 매일 1km 이상 걸리기! 목표를 달성하면 나이키 신발 10% 할인 쿠폰을 드립니다. 🏃
- 예시: 1주일 동안 6.5-7.5시간 수면! 목표를 달성하면 대안 10% 할인 쿠폰을 드립니다. 😴
- 곧 정식 서비스로 만나볼 수 있습니다. 조금만 기다려주세요. 🕒

## 출력 인터페이스

- 1) 사용자 예측 건강 등급 -> 사용자가 입력한 데이터를 바탕으로 머신러닝 모델을 통해 얻은 등급 표시
- 2) 건강 그래프 -> 사용자의 건강 그래프(하늘색)와 사용자가 속한 연령대의 평균 건강 그래프(주황색)가 표시
- 3) 현재 사용자의 건강 상태를 바탕으로 건강 팁들을 출력

## 3-1. 웹 서비스 개발 과정

```
# 건강 등급 예측 및 시각화 함수 (generate_health_tips 호출 수정)
def predict_health_score(first_name, last_name, age, height, weight, smoking, exercise, diet, sleep, alcohol, health_issue="", email=""):
    # 입력값 검증

    # BMI 계산
    bmi_val = weight / ((height / 100) ** 2)

    # 점수 계산
    bmi_score = BMI(bmi_val)
    exercise_score = EXERCISE(exercise)
    diet_score = DIET(diet)
    sleep_score = SLEEP(sleep)
    alcohol_score = ALCOHOL(alcohol)
    smoking_score = SMOKING(smoking)

    # 연령대 평균 계산
    age_group = get_age_group(age)
    age_group_health_class = df_clean[['Age', 'New_Health_Class']]
    try:
        age_group_avg = df_clean[df_clean['Age'].between(age_group, age_group + 9)][['Exercise_Score', 'Diet_Score', 'Sleep_Score',
                                                                                       'Alcohol_Score', 'Age', 'Exercise_Frequency',
                                                                                       'Diet_Quality', 'Sleep_Hours',
                                                                                       'Alcohol_Consumption']].mean()
```

### 주요 함수(predict\_health\_score)

- 1) 함수 정의, 매개변수는 전부 입력 인터페이스에서 입력받은 데이터
- 2) 입력 인터페이스의 수치형 데이터들을 자체 기준에 따라 알맞는 점수로 변환

## 3-1. 웹 서비스 개발 과정

```
# 연령대 평균 계산
age_group = get_age_group(age)
age_group_health_class = df_clean[['Age', 'New_Health_Class']]
try:
    age_group_avg = df_clean[df_clean['Age'].between(age_group, age_group + 9)][['Exercise_Score', 'Diet_Score', 'Sleep_Score',
                                                                                   'Alcohol_Score', 'Age', 'Exercise_Frequency',
                                                                                   'Diet_Quality', 'Sleep_Hours',
                                                                                   'Alcohol_Consumption']].mean()

# 각 값 정규화 (0~100 범위 조정, 시각화용)
exercise_scaled = ((EXERCISE(exercise)-2)/8) * 100 # 운동 (0~10점 → 0~100)
diet_scaled = DIET(diet) * 10 # 식단 (0~10점 → 0~100)
sleep_scaled = ((SLEEP(sleep)-1)/9) * 100 # 수면 시간 (0~10점 → 0~100)
alcohol_scaled = ((ALCOHOL(alcohol)-1)/9) * 100 # 음주량 (0~10점 → 0~100)

avg_exercise_scaled = age_group_avg['Exercise_Score'] * 10
avg_diet_scaled = age_group_avg['Diet_Score'] * 10
avg_sleep_scaled = age_group_avg['Sleep_Score'] * 10
avg_alcohol_scaled = age_group_avg['Alcohol_Score'] * 10

user_data = [exercise_scaled, diet_scaled, sleep_scaled, alcohol_scaled]
avg_data = [avg_exercise_scaled, avg_diet_scaled, avg_sleep_scaled, avg_alcohol_scaled]

# 시각화 레이블용 단위 설정 (사용자 입력 단위로 표시)
user_inputs = [exercise, diet, sleep, alcohol]
avg_inputs = [
    age_group_avg['Exercise_Frequency'], # 회/주
    age_group_avg['Diet_Quality'], # 0~100 스케일
    age_group_avg['Sleep_Hours'], # 시간
    age_group_avg['Alcohol_Consumption'] # 회/주
]
```

### 주요 함수(predict\_health\_score)

3) 사용자가 속한 연령대의 평균값 계산

4) 그래프를 그리기 위해 수치들 0~100 사이 값을 가지도록 조정

5) 그래프 작성 함수에서 사용할 매개변수 생성

## 3-1. 웹 서비스 개발 과정

```
# 건강 관리 팁 리포트 생성 (Base64 이미지 전달)
health_tips = generate_health_tips(health_class, health_issue, exercise, diet, sleep, alcohol,
                                   smoking, bmi_score, exercise_score, diet_score, sleep_score,
                                   alcohol_score, smoking_score, age_group_avg, first_name, base64_image=base64_image)

# 이메일 전송 (입력된 경우)
if email and email.strip():
    try:
        email_result = send_email(email, health_class, health_tips, fig, first_name, last_name)
        if email_result is True:
            health_tips += "\n\n• 이메일 전송 결과: 리포트가 성공적으로 발송되었습니다! 📧"
        else:
            health_tips += f"\n\n• 이메일 전송 결과: {email_result} 📧"
    except Exception as e:
        health_tips += f"\n\n• 이메일 전송 중 예외 발생: {str(e)} 📧"

return health_tips
```

### 주요 함수(predict\_health\_score)

6) 출력 인터페이스에 들어갈 값을 생성하기 위한 health\_tips 생성

7) 이메일 전송관련 메시지 출력

## 3-1. 웹 서비스 개발 과정

```
# HealthSquare 시각화 함수 (Base64 인코딩 추가, units 정의)
def plot_health_square(user_values, avg_values, health_class, user_inputs, avg_inputs):
    center_x, center_y = 75, 75
    base_size = 30

    def get_corners(values):
        exercise, diet, sleep, alcohol = values
        return [
            ((center_x - 5) - base_size * (exercise / 100), (center_y + 5) + base_size * (exercise / 100)), # 운동
            ((center_x + 5) + base_size * (diet / 100), (center_y + 5) + base_size * (diet / 100)), # 식단
            ((center_x + 5) + base_size * (sleep / 100), (center_y - 5) - base_size * (sleep / 100)), # 수면
            ((center_x - 5) - base_size * (alcohol / 100), (center_y - 5) - base_size * (alcohol / 100)) # 음주
        ]

    user_corners = get_corners(user_values)
    avg_corners = get_corners(avg_values)

    fig, ax = plt.subplots(figsize=(5, 5))
    user_polygon = plt.Polygon(user_corners, fill=True, color='skyblue',
                               alpha=0.6, label="Your Health")
    ax.add_patch(user_polygon)
    avg_polygon = plt.Polygon(avg_corners, fill=True, color='orange',
                              alpha=0.4, label=f"Average Health")
    ax.add_patch(avg_polygon)

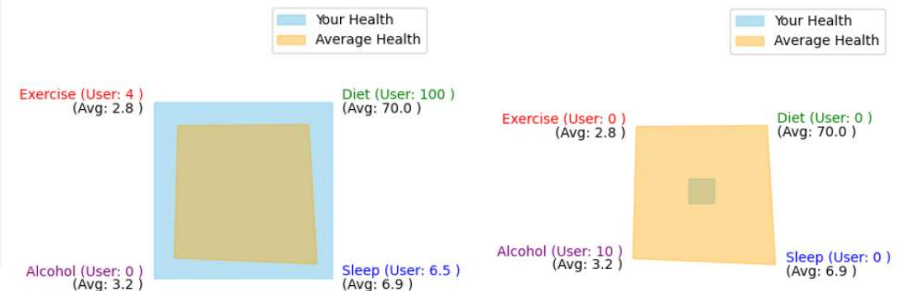
    ax.set_xlim(0, 150)
    ax.set_ylim(0, 150)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_frame_on(False)
```

### 주요 함수(plot\_health\_square)

1) x축 y축 범위를 지정한 뒤(150, 150) 정 가운데에 그래프가 그려지도록 center 지정

2) 점수에 따라 그래프가 과하게 커지거나 작아지는 것을 방지하기 위한 base\_size 설정

3) 사용자 그래프의 꼭짓점, 평균 그래프의 꼭짓점 지정



## 3-1. 웹 서비스 개발 과정

```
label_offset = 4
labels = ["Exercise", "Diet", "Sleep", "Alcohol"]
colors = ["red", "green", "blue", "purple"]
units = ["", "", "", ""]

for i, (user_val, avg_val, user_input, avg_input) in enumerate(zip(user_values, avg_values, user_inputs, avg_inputs)):
    if user_val >= avg_val:
        label_x, label_y = user_corners[i]
    else:
        label_x, label_y = avg_corners[i]

    ha = 'right' if i in [0, 3] else 'left'
    va = 'bottom' if i in [0, 1] else 'top'

    ax.text(label_x + (label_offset if ha == 'left' else -label_offset),
            label_y,
            f"{labels[i]} (User: {user_input} {units[i]})",
            fontsize=10, ha=ha, va='bottom', color=colors[i])
    ax.text(label_x + (label_offset if ha == 'left' else -label_offset),
            label_y,
            f"(Avg: {avg_input:.1f} {units[i]})",
            fontsize=10, ha=ha, va='top', color='black')

ax.legend()

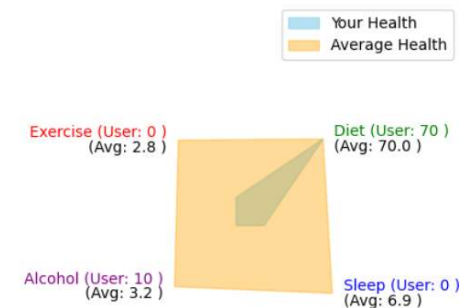
# 이미지를 Base64로 인코딩
buffer = BytesIO()
fig.savefig(buffer, format="png", bbox_inches='tight')
buffer.seek(0)
image_png = buffer.getvalue()
base64_string = base64.b64encode(image_png).decode('utf-8')
plt.close(fig)

return fig, base64_string
```

### 주요 함수(plot\_health\_square)

4) 그래프의 각 꼭짓점에 사용자 입력 데이터 라벨, 평균 라벨을 출력. 라벨끼리 겹치지 않게 하고 그래프와도 겹치지 않게 출력

5) 그래프를 메모리에 저장하여 처리 속도를 향상



## 3-2. 웹서비스 시연



헬스케어 접속 QR코드

## 4. 프로젝트 결과

4-1. 시사점

4-2. 개선점



## 4-1. 시사점

1. 알고리즘 다양성과 성능 최적화
2. 데이터셋 왜곡 및 편향

## 4-2. 개선점

1. 웹서비스를 통한 데이터 수집
2. 사용자 참여 유도 전략 필요성