헬스퀘어 : 사각형 건강남녀가 주목받는 시대

팀명 : 잘살게해주호
팀원 : 권수현, 김채영, 오주호, 지준희

1. 개요

1-1. 주제 선정 배경

현대사회에서 가장 이상적인 인간상으로 '육각형 인간'이 떠오르고 있다. 육각형 인간이란 외모도 준수하고, 성격도 모난 데 없으며, 집안과 직업이 탄탄하고, 고학력에 자산을 많이 축적한 사람을 의미한다. 부족한 것 하나 없이 모든 면에서 평균 이상을 갖추고 있기 때문에 연애나 결혼 상대로도 최고의 이상형에 꼽힌다. 즉, 육각형 인간이란 하나라도 타인보다 뒤쳐지고 싶지 않은 대한민국 사회의 욕구를 반영하며 자기관리를 위한 프레임으로도 많이 활용되고 있다.

육각형 인간을 이루고 있는 항목 가운데 자산이란 요소는 매력 자산, 경제 자산, 건강 자산 세 가지로 나뉜다. 외모와 성격이 매력 자산을 구성하는 요소라면 집안, 직업, 학력은 경제 자산을 구성하는 요소라고 할 수 있다. 반면 건강 자산에 해당하는 요소는 한 가지도 없다. 아마도 건강을 경제적 가치로 환산하는 게 어렵기 때문일 것이다. 하지만 건강이 악화되면 사람은 외적 매력도 잃게 되고 그동안 축적했던 경제적 기반이 급속도로 무너질 수도 있다. 따라서 건강 자산을 먼저 튼튼하게 구축해야 매력 자산과 경제 자산이 유의미하기 때문에 예방이 중요해지고 있다.

잘살게 해주호 팀에서는 육각형 인간에서 착안한 컨셉을 헬스케어에 접목하여 헬스퀘어 (HealthSquare : Healthcare + Square) 서비스를 출시했다. 건강점수에 영향을 미치는 대표적인 네 가지 항목(식사, 수면, 운동, 음주)의 데이터를 기반으로 본인의 건강점수를 확인하고, 비슷한 연령대의 다른 사람들과 비교할 수 있다. 또한 건강점수를 높이기 위한 팁을 얻는 동시에 목표를 달성하면 보상까지 받는 시스템을 통해 사용자는 하나의 통합 서비스에서 꾸준하고 즐겁게 건강 관리를 할 수 있다. 헬스퀘어와 함께라면 누구든지 꽉 찬 사각형 건강남녀로 거듭날 수 있다.

1-2. 진행 과정

1) Kaggle에서 수집한 건강 데이터를 분석하여 결측치와 이상치를 처리한다.

2) 수치형 변수를 범주화하여 단위를 통일하고 변수 간의 상관관계를 분석한다.

3) 여러 가지 머신러닝 알고리즘의 성능을 비교하여 최적의 분류 모델을 선정한다.

4) Gradio를 통해 일반 사용자를 대상으로 제공되는 헬스케어 웹 서비스를 개발한다.

2. 개발환경

2-1. 운영체제 및 파이썬 버전

- Windows 10 Pro (64비트)
- Python 3.11.11 인터프리터

2-2. 라이브러리

1) 데이터분석

- Pandas (2.2.2), Numpy (1.26.4), Matplotlib(3.10.0), Seaborn(0.13.2), sklearn(

2) 머신러닝

- KNN(1.6.1),RandomForest(1.6.1), LogisticRegression(1.6.1), XGBoost(2.1.4), LightGBM(4.5.0), CatBoost(1.2.7)

3. 일정

3-1. 워크로드

1) 프로젝트 기획 : 주제 선정, 데이터 수집

2) 데이터분석 : 전처리, 시각화

3) 머신러닝 : 모델 학습 및 평가, 최적 모델 도출

4) 웹 서비스 : 페이지 구현, 버그 수정

3-2. 타임라인

Week	Week 1					Week 2			
Date	2.24 (월)	25 (화)	26 (수)	27 (목)	28 (금)	3.4 (화)	5 (수)	6 (목)	7 (금)
주제선정									
데이터 수집									
기안서 작성									
데이터분석									
머신러닝									
웹 서비스									
개발서 작성									
발표자료 작성									
발표									

4. 데이터셋

4-1. 소개

1) 제목 : Health and Lifestyle Data for Regression
   * 본 프로젝트에서는 파생변수(New_Health_Score)를 생성하여 분류(Categorize)로 진행

2) 출처 : Kaggle.com
   * URL : https://www.kaggle.com/datasets/pratikyuvrajchougule/health-and-lifestyle-data-for-regression

4-2. 변수

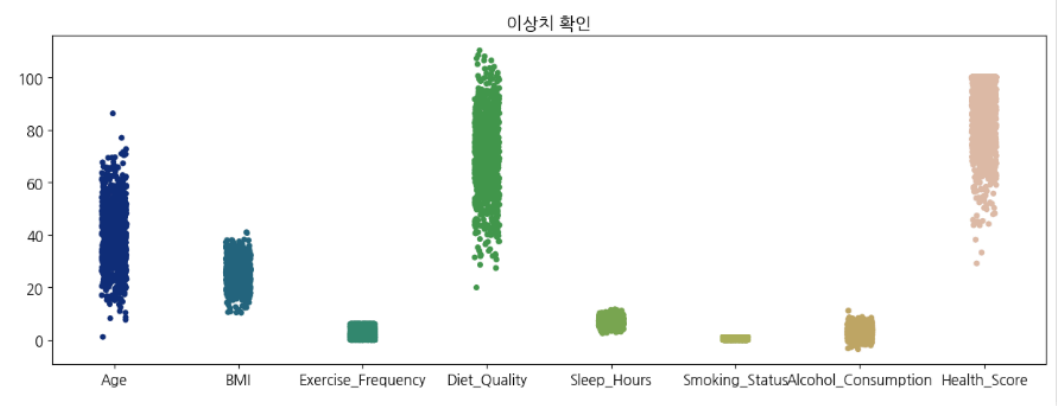
1) 독립변수(X)
- Age : 나이
- BMI : 몸무게(kg) / (키(m)^2))
- Exercise_Frequency : 운동횟수(per week, categorical, 0-7)
- Diet_Quality : 식사질(continuous, 0-100)
- Sleep_Hours : 수면시간(per day, continuous, 0-12)
- Smoking_Status : 흡연여부(binary, 0 = Non-smoker, 1 = Smoker)
- Alcohol_Consumption : 음주량(per week, continuous, 0-12)

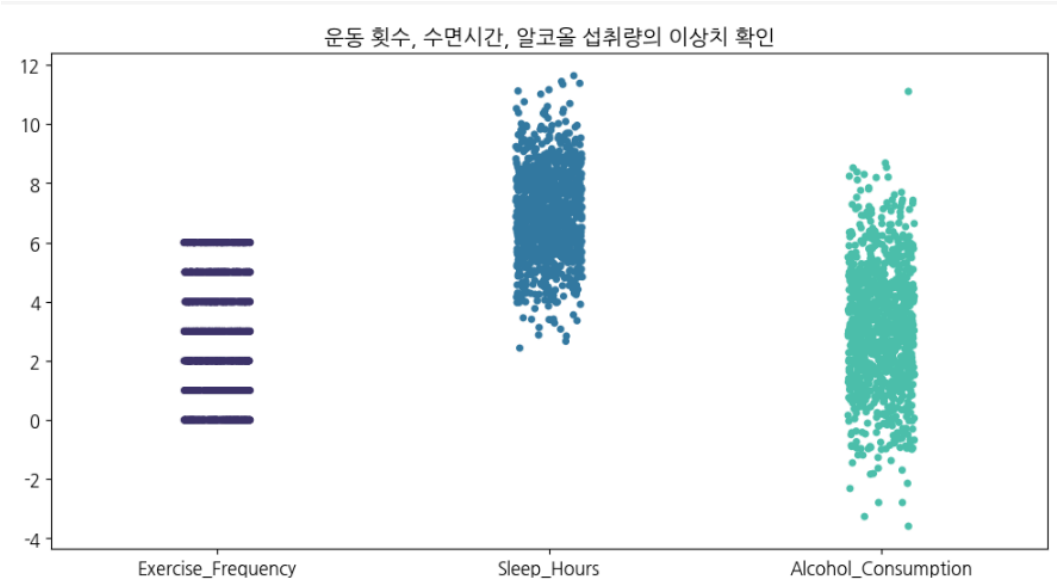
2) 종속변수(y) : Target
- Health_Score : 건강점수(continuous, 0-100)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 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)
```

5. 데이터 전처리 및 머신러닝 구현

5-1. EDA





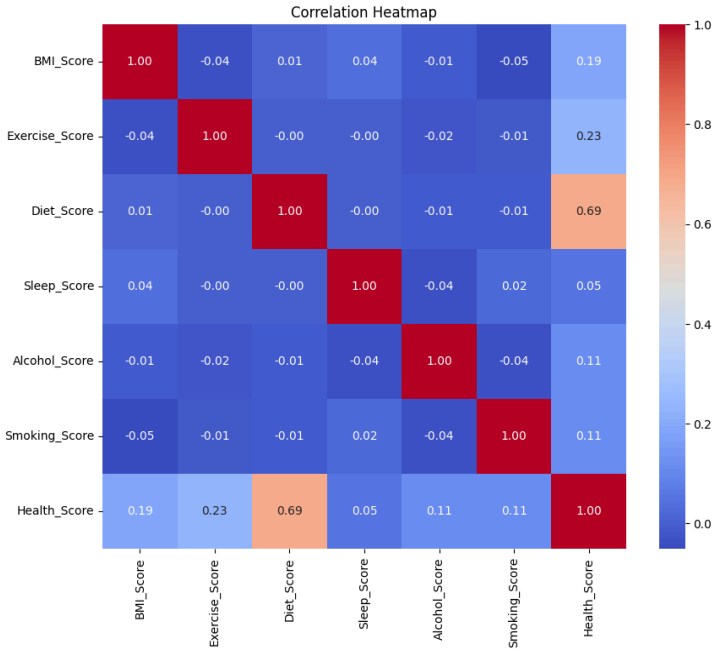
1) 결측치 없음
2) 이상치가 있는 열 3개 전처리
   - Age: 1살부터 89살까지 연령 데이터 존재
     => 신생아 및 청소년과 성인의 BMI 기준이 다르기 때문에 20세 미만 37명 제거
   - Diet_Quality: 만점(100점)을 초과하는 이상치 n개 존재
     => 100점으로 변환
   - Alcohol_Consumption: 일주일을 기준으로 섭취한 알코올의 횟수가 음수값이 존재
     => 절댓값 처리

5-2. 독립변수 범주화 후 회귀 모델 적용

```python
import numpy as np

def categorize_bmi(bmi):
    if 10 <= bmi < 11: return 1
    elif 11 <= bmi < 12: return 2
    elif 12 <= bmi < 13: return 3
    elif 13 <= bmi < 14: return 4
    elif 14 <= bmi < 15: return 5
    elif 15 <= bmi < 16: return 6
    elif 16 <= bmi < 17: return 7
    elif 17 <= bmi < 18: return 8
    elif 18 <= bmi < 18.5: return 9
    elif 18.5 <= bmi < 25: return 10
    elif 25 <= bmi < 27.5: return 9
    elif 27.5 <= bmi < 30: return 8
    elif 30 <= bmi < 32.5: return 7
    elif 32.5 <= bmi < 35: return 6
    elif 35 <= bmi < 36: return 5
    elif 36 <= bmi < 37: return 4
    elif 37 <= bmi < 38: return 3
    elif 38 <= bmi < 39: return 2
    else:
        return 1

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

- 변수들의 단위가 전부 달라 스케일링 개념으로 모든 변수를 10점을 만점으로 범주화
  ex) BMI 정상 범위는 18.5~25. 숫자가 위로 커지든 아래로 커지든 과체중이나 저체중으로 인해 건강이 안 좋다는 것을 의미하므로 정상 범위에 해당하는 값을 만점으로 설정하고 양 옆으로 갈수록 점수가 감소하는 방식으로 범주화를 진행. (정규분포에 가까운 형태)



- 결과 : Diet_Quality의 상관관계만 높기 때문에 건강지표 산출이 객관적이지 않음. (0.69)
- 회귀 모델을 적용하여 Health_Score를 예측하면 Diet_Quality의 영향이 과도하게

계상됨.

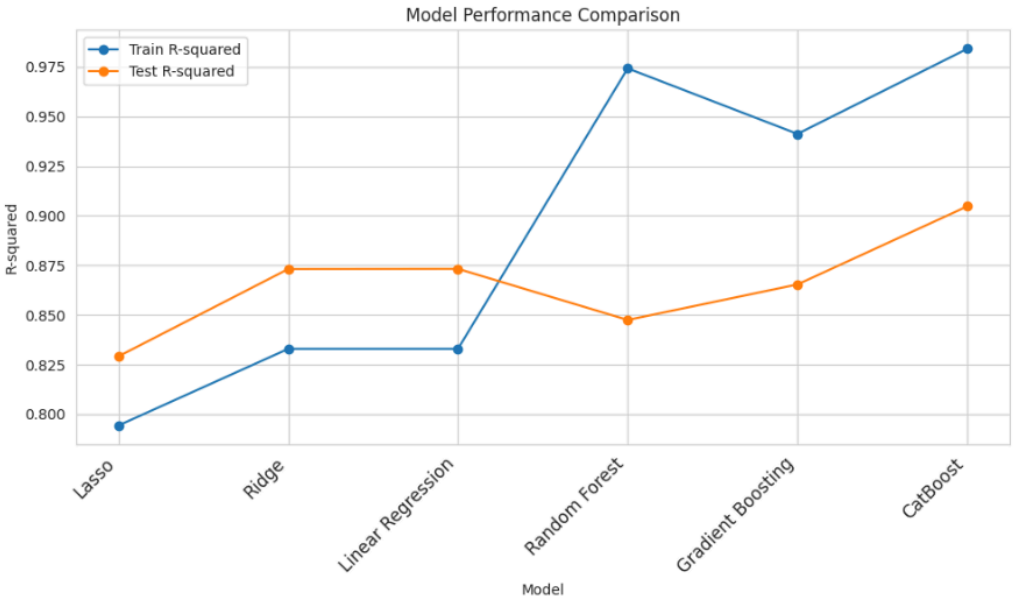
5-3. 특성 공학을 활용한 다항회귀 모델 적용

- 건강점수(y) 예측에 필요한 요소들(X : 식단, 수면시간, 운동, 알코올 섭취량, 흡연 여부)이 서로 다른 가중치를 가져야 한다고 판단.
- 아래의 식과 같은 원리로 다중회귀 모델을 적용.
  y = ax1(식사) + bx2(수면) + cx3(운동) + dx4(음주) + ex5(흡연)
- 머신이 독립변수(X)와 종속변수(y)를 비선형적인 관계로 예측할 가능성을 고려하여 일차항(x1, x2, x3, x4, x5)을 고차항과 교차항으로 확장하는 특성 공학을 활용.
- 특성 공학을 통해 확장된 변수를 바탕으로 다항회귀 모델 적용.
  * 특성 공학으로 인해 발생하는 다중공선성 문제는 일부 감안하고 진행.

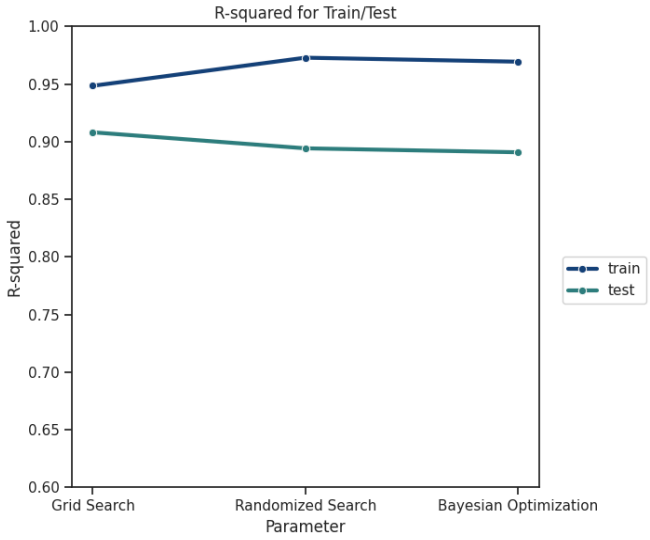
- 결과 : 준수한 성능을 보이나, 산점도 그래프를 통해 확인한 결과 선형적 관계를 보일 수도 있다고 판단하여 다른 방법 시도.

5-4. 스케일링 및 하이퍼파라미터 튜닝

- Diet_Quality의 절대적인 요소값이 다른 요소에 비해 크기 때문에 Target(y)과 상관관계가 높았을 것이라고 추정하여 스케일링 진행. (StandardScaling, MinMaxScaling)
- 스케일링을 진행한 후에도 Target(y)과의 상관관계가 높으므로 Diet_Quality 요소값 자체의 문제는 아니라고 판단하여 모델 학습 진행.
- 최적의 모델인 CatBoost에서도 과대적합 발생. (Train_Score : 0.97, Test_Score : 0.90)



- 결정계수가 가장 높은 CatBoost 모델을 쓰되, 하이퍼파라미터 튜닝으로 과대적합 해소 (**Grid Search CV, Randomized Search CV, Bayesian Optimization**)

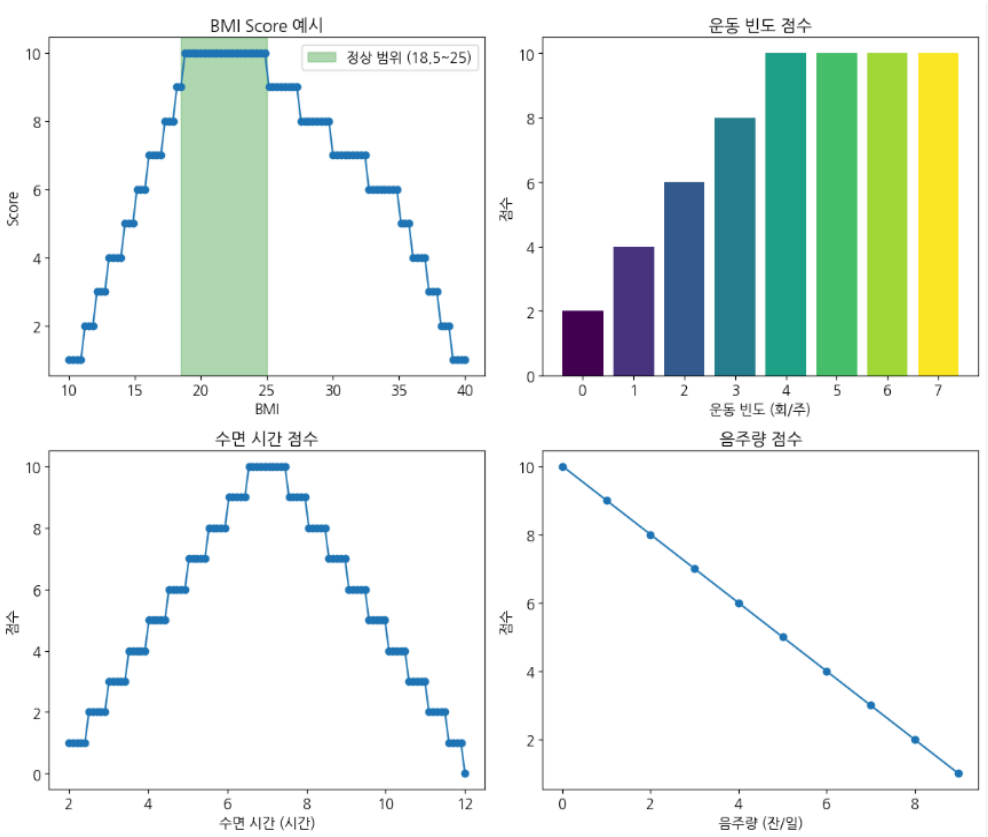


- 결과: Grid Search 를 통해 과대적합 해소(train: 0.948/test:0.90)
- 한계점: 원본 데이터셋에서 **Target(Health_Score)**의 편중이 과도하기 때문에 머신러닝 모델을 학습시켜도 합리적인 결과값을 도출하는 것이 불가능.
  1) 객관성과 합리성이 결여된 건강점수
  => Health_Score 산출 시 Diet_Quality가 과하게 높은 가중치를 갖기 때문에 모델상으로는 식단 관리만 잘하면 수면시간, 운동, 알코올 섭취량, 흡연 여부에 관계없이 건강점수가 우수할 것이라고 예측하는 비상식적인 결과가 도출됨.
  2) 스케일링으로 인한 입력값(실제값)과 출력값(표준값)의 괴리
  => Health_Score를 계산하는 모델은 스케일링된 표준값(-1~1 값)인 데 반해 사용자가 입력하는 건강지표는 실제값(Diet 80점,Exercise 4회, Sleep 6시간, Alcohol 2잔 등)이므로 2 이상 값을 입력하면 건강점수의 변동이 생기지 않음.

5-5. 신규 Target 생성

- 과학적인 건강 기준에 근거하여 신규 Target(New_Health_Score)을 생성.
- 신규 Target은 Diet, Smoking, BMI, Exercise, Sleep, Alcohol로 구성.
- New_Health_Score를 계산하기 위해 각 요소들의 만점 단위를 10점으로 통일.
- **New_Health_Score**
  = (BMI_Score * 2 + Exercise_Score * 2 + Diet_Score * 2 + Sleep_Score * 2 + Alcohol_Score * 2) + Smoking_Score
  * New_Health_Score의 만점이 100이 되도록 각 Score에 2를 곱해 합산
  * Smoking_Score는 흡연자의 경우 -5점을 감점하는 형태로 반영
  * 과학 도메인 지식을 활용하여 각 Score의 가중치 조정 가능

1) Diet_Quality : 10으로 나누어 100점 만점을 10점 만점으로 변환
2) Smoking_Status : 비흡연자(0)는 0점, 흡연자(1)는 -5점 부여.
3) BMI, Exercise_Frequency, Sleep_Hours, Alcohol_Consumption
   : 각 요소마다 정상 범위에 해당하는 점수를 10점 만점으로 하여 정상 범위에서
   벗어날수록 감점하는 방식으로 Score column을 생성.



* BMI : 18.5 <= x < 25 (정상, 10점)
    10 <= x < 11 1점, 11 <= x < 12 2점, 12 <= x < 13 3점,
    13 <= x <14 4점, 14 <= x < 15 5점, 15 <= x <16 6점,
    16 <= x < 17 7점, 17 <= x < 18 8점, 18 <= x < 18.5 9점,
    25 <= x < 27.5 9점, 27.5 <= x < 30 8점, 30 <= x < 32.5 7점, 32.5 <=
    x < 35 6점, 35 <= x < 36 5점, 36 <= x < 37 4점,
    37 <= x < 38 3점, 38 <= x < 39 2점, 39 <= x 1점
* Exercise_Frequency : x >= 4 (정상, 10점)
    X == 0 2점, x == 1 4점, x == 2 6점, x == 3 8점
* Sleep_Hours : 6.5 <= x < 7.5 (정상, 10점)
    0 <= x < 2.5 1점, 2.5 <= x < 3 2점, 3 <= x < 3.5 3점,
    3.5 <= x < 4 4점, 4 <= x < 4.5 5점, 4.5 <= x < 5 6점,
    5 <= x < 5.5 7점, 5.5 <= x < 6 8점, 6 <= x < 6.5 9점,
    7.5 <= x < 8 9점, 8 <= x < 8.5 8점, 8.5 <= x < 9 7점,
    9 <= x < 9.5 6점, 9.5 <= x < 10 5점, 10 <= x < 10.5 4점,
    10.5 <= x < 11 3점, 11 <= x < 11.5 2점, 11.5 <= x < 12 1점
* Alcohol_Consumption : 0 <= x < 1 (정상, 10점)
    1 <= x < 2 9점, 2 <= x < 3 8점, 3 <= x <4 7점,

4 <= x < 5 6점, 5 <= x < 6 5점, 6 <= x < 7 4점,
7 <= x < 8 3점, 8 <= x < 9 2점, 9 <= x 1점

- 공식을 통해 도출한 New_Health_Score를 회귀 머신러닝 모델로 예측하는 것은 무의미하기 때문에 New_Health_Score 점수를 10점 단위로 나누어 구간 추정, 즉 회귀 대신 분류 모델을 구축하는 방향으로 재설정

- 최종 **Target : Health_Class**
  Health_Class : New_Health_Score에 따라 6단계(A등급~F등급)로 분류
  90 <= NHS A등급, 80 <= NHS < 90 B등급, 70 <= NHS < 80 C등급,
  60 <= NHS < 70 D등급, 50 <= NHS < 60 E등급, NHS < 50 F등급

5-6. 분류 알고리즘 성능 비교

- 하이퍼파라미터 튜닝 이전 모델 성능 비교

No.	Model	Train Accuracy	Test Accuracy	Fitting
1	KNN	0.8403	0.7668	Overfitting
2	RandomForest	1.0000	0.7824	Overfitting
3	XGBoost	1.0000	0.7668	Overfitting
4	LightGBM	1.0000	0.7565	Overfitting
5	CatBoost	1.0000	0.8394	Overfitting
6	**LogisticRegression**	**0.9597**	**0.9482**	**Fitting**
Best_Model : LogisticRegression				

- 하이퍼파라미터 튜닝 이후 모델 성능 비교

No.	Model	Hyper-Parameter Tuning	Train Accuracy	Test Accuracy	Fitting
1	KNN	n_neighbors: 5 weights: distance	0.8403	0.7668	Overfitting
2	RandomForest	max_depth: None n_estimators: 100	1.0000	0.7824	Overfitting
3	XGBoost	learning_rate: 0.1 n_estimators: 100	1.0000	0.7668	Overfitting
4	LightGBM	learning_rate: 0.2 n_estimators: 100	1.0000	0.7565	Overfitting
5	CatBoost	iterations: 200 learning_rate: 0.1	1.0000	0.8394	Overfitting
6	**LogisticRegression**	max_iter: 1000 class_weight=balanced	**0.9455**	**0.9689**	**Fitting**
Best_Model : LogisticRegression					

=== 최적 모델===
Model: LogisticRegression (등급 가중치 조정)
* 등급 가중치 조정 : 건강 등급 간 샘플 차이가 클 때 각 클래스에 다른 가중치를 부여하는 방식
Train Accuracy: 0.9455
Test Accuracy: 0.9689
Test Classification Report:

	precision	recall	f1-score	support
A	0.86	1.00	0.92	6
B	0.98	0.98	0.98	42
C	0.99	0.97	0.98	75
D	0.98	0.95	0.96	55
E	0.88	1.00	0.93	14
F	1.00	1.00	1.00	1
accuracy			0.97	193
macro avg	0.95	0.98	0.96	193
weighted avg	0.97	0.97	0.97	193

6. 웹 서비스

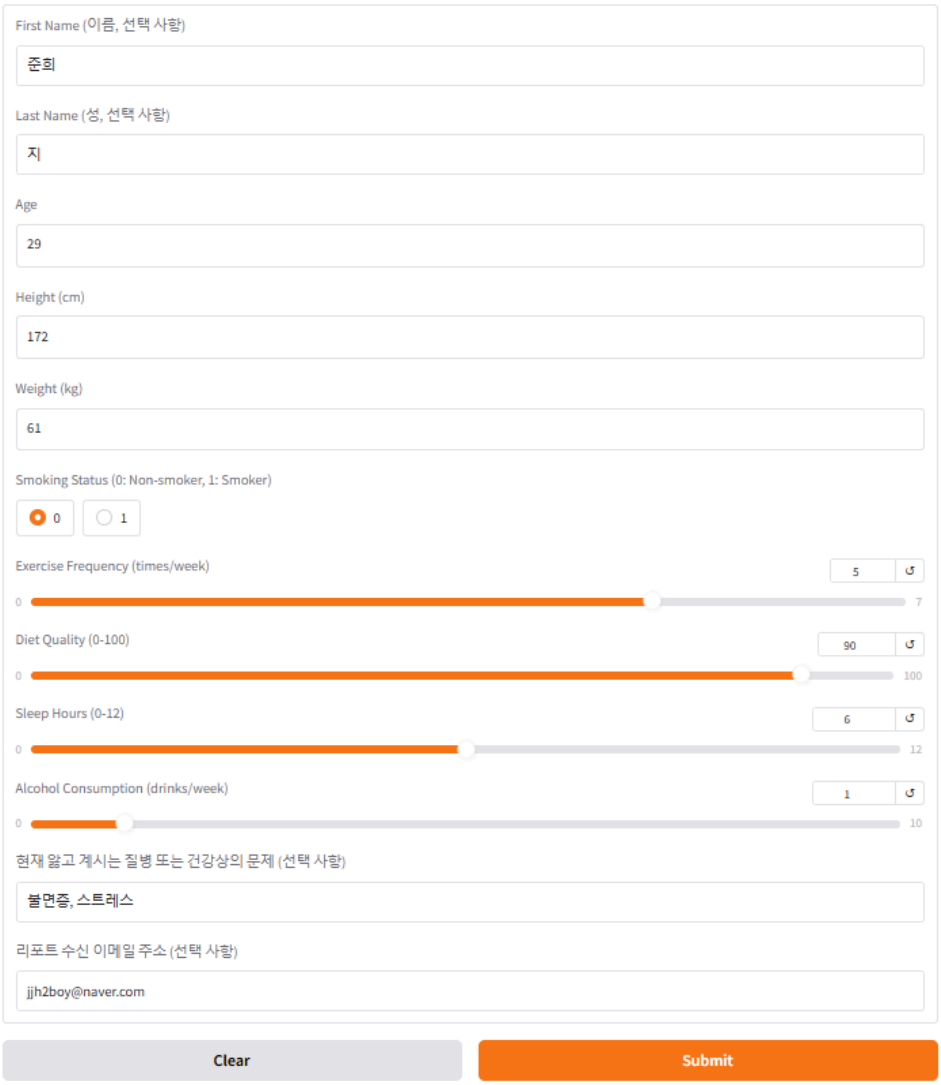
6-1. 웹 서비스 개발 라이브러리

- **Gradio**
  : Python에서 웹 인터페이스를 만들고, 머신러닝 모델 및 데이터 분석 기능을 배포할 수 있는 쉽고 간편한 라이브러리인 Gradio를 선정

6-2. 웹 서비스 UI 및 구현 코드

1) 입력 부분

< 유저 정보 입력 인터페이스 >



< 유저 정보 입력 인터페이스 구현 코드 >

```
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 앓고 계시는 질병 또는 건강상의 문제 (선택 사항)", placeholder="예: 불면증, 다이어트 중", value=""),
    gr.Textbox(label="리포트 수신 이메일 주소 (선택 사항)", placeholder="예: example@email.com", value="")
],
```

- 머신러닝 과정에서 최종 선정된 모델을 기반으로 Gradio에서 실행시킬 함수를 **predict_health_score**라고 정의하고, 유저가 입력한 건강 지표는 차례대로 predict_health_score의 매개변수로 투입.

- Age : 유저가 입력한 값을 바탕으로 해당 연령대의 Average_Score를 계산
- Height, Weight : 유저가 입력한 값을 바탕으로 BMI_Score를 계산
- Smoking_Status : 유저가 선택한 값을 바탕으로 Smoking_Score를 계산
- Exercise_Frequency, Diet_Quality, Sleep_Hours, Alcohol_Consumption
  : 유저가 입력한 값을 바탕으로 각각의 Score를 계산
  => 내부 알고리즘에서는 건강점수(New_Health_Score)를 계산하지만 실제로 유저가 확인하는 화면에는 건강등급(Health_Class)으로 변환하여 제공
  => 유저의 건강 사각형과 동일 연령대 평균 건강 사각형을 겹쳐서 보여주는 방식을 통해 유저가 건강점수를 개선할 수 있는 영역을 시각적으로 표현

- Health_Issues(optional) : 유저가 앓고 있는 질병 또는 건강상의 문제
- E-mail_Address(optional) : 유저가 건강 리포트를 받을 이메일 주소
  => 유저가 건강점수와 건강등급을 개선하기 위해 취할 수 있는 조치와 함께 도움이 되는 제품 또는 서비스를 개인 맞춤형 리포트로 발송
  => 유저가 건강 관련 목표를 설정하고 달성할 수 있도록 유도함으로써 헬스케어 서비스를 지속적으로 이용하고, 추가 데이터(성별, 국적 등)와 시계열 데이터를 축적

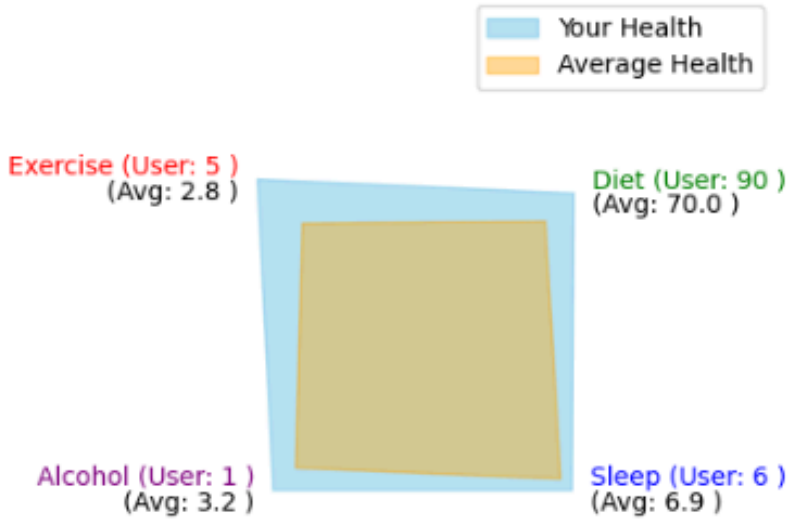
2) 출력 부분

< 헬스케어 건강 리포트 출력 인터페이스 **>**

📋 Check - 건강 상태 확인 📋

• 당신의 건강 등급: A

📊 Compare - 연령대 비교 📊



• 동일 연령대(20.0대)와 비교:

- 운동 빈도: 당신 5회/주 (평균: 2.8회/주, +2.2회) 🚴
- 식단 퀄리티: 당신 90 (평균: 70.0, +20.0) 🥗
- 수면 시간: 당신 6시간 (평균: 6.9시간, -0.9시간) 🌙
- 음주량: 당신 1회/주 (평균: 3.2회/주, -2.2회) 🍷

• 식단 퀄리티가 매우 좋으시네요! 현재 식단을 유지하시면서, 간식으로 견과류나 과일을 추가해보세요. 🍎
• 수면 시간이 짧은 편입니다. 하루 7~8시간 수면을 목표로 하세요. 취침 전 따뜻한 카모마일 차를 마시면 숙면에 도움이 됩니다. 🍵

💡 Coaching - 건강 개선 팁 💡

- 준희님, 불면증, 스트레스로 인해 어려움을 겪고 계시군요. 몇 가지 구체적인 조언을 드릴게요. 💡
- 스트레스를 받으실 때는 균형 잡힌 식단과 규칙적인 수면이 중요합니다. 💡
- 스트레스 완화에 도움이 되는 영양제로 **테아닌(L-theanine)**을 추천드립니다. 💊
- **테아닌**은 신경을 안정시키고 긴장을 완화하는 효과가 있어 스트레스 관리에 좋습니다. 녹차나 홍차에 포함되어 있지만, 영양제 형태로 섭취하는 것이 더 간편합니다. 🍵
- **테아닌**에 관심이 있으시다면 'Now Foods'의 'L-Theanine 100mg'를 추천드립니다. 약 1달 분량(60정)을 약 15,000원에 구매하실 수 있습니다. 🛒
- 또한, 스트레스를 완화하기 위해 가벼운 운동이 큰 도움이 됩니다. 날씨가 좋은 날 집 근처 공원에서 인터벌 러닝을 해보세요. 🚴
- 인터벌 러닝은 3분 동안 빠르게 달린 후 2분 동안 천천히 걷는 과정을 20~30분간 반복하는 방식입니다. 이렇게 하면 러닝에 집중하면서 스트레스를 해소하는 데 도움이 됩니다. ⏱️
- 수면의 질을 높이기 위해 규칙적인 수면 패턴을 유지하는 것이 중요합니다. 수면 개선에 도움이 되는 영양제로 **멜라토닌**을 추천드립니다. 💊
- **멜라토닌**은 수면 주기를 조절하는 데 도움을 주며, 불면증 완화에 효과적입니다. 체리, 포도, 월넛에 포함되어 있지만, 영양제로 섭취하면 더 간편합니다. 🍒
- **멜라토닌** 영양제로는 'Natrol'의 'Melatonin 3mg'를 추천드립니다. 약 1달 분량(60정)을 약 15,000원에 구매하실 수 있습니다. 🛒
- 수면 개선을 위해 취침 전 가벼운 스트레칭이나 명상도 추천드립니다. '다리 흔들기' 스트레칭(누워서 다리를 1분간 흔들기)을 해보세요. 긴장을 풀고 숙면에 도움이 됩니다. 🌙

🎁 Coupon - 목표 설정 및 쿠폰 제공 (베타 서비스 준비 중) 🎁

- 준희님, 목표를 설정하고 달성하면 쿠폰을 드립니다! 현재 베타 서비스 준비 중입니다. 🎯
- 예시: '1달 동안 매일 1km 이상 달리기' 목표를 달성하면 **나이키 신발 10% 할인 쿠폰**을 드립니다! 🚴
- 예시: '1주일 동안 6.5~7.5시간 수면' 목표를 달성하면 **테아닌 10% 할인 쿠폰**을 드립니다! 🌙
- 곧 정식 서비스로 만나보실 수 있습니다. 조금만 기다려주세요! 😊

- 이메일 전송 결과: 리포트가 성공적으로 발송되었습니다! 👍

< 헬스케어 건강 리포트 출력 인터페이스 구현 코드 >

```
outputs=[
    gr.Markdown(label="건강 관리 팁 리포트") # Plot이 Markdown 내에 포함될 것으로 예상
],
```

- predict_health_score 함수 값을 return하여 유저의 건강 리포트를 출력
- gr.Markdown이라는 컴포넌트를 통해 자동으로 리포트 레이아웃 조정
- 유저의 이메일로 전송되는 리포트는 html 형식으로 변환하여 제공

3) 주요 함수 코드

<건강지표 정규화 함수 코드 >

```
# 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
]

try:
    fig, base64_image = plot_health_square(user_data, avg_data, health_class,
                                           user_inputs, avg_inputs)
except Exception as e:
    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)
    return health_class, None, f"Error in plotting HealthSquare: {str(e)}₩n₩n건강 관리 팁 리포트:₩n{health_tips}"
```

- 유저의 데이터와 연령대 평균 데이터가 0~100점 사이가 되도록 조정
- 그래프 출력 함수의 매개변수에 넣기 위해 user_data와 avg_data 변수로 각각 저장

< 건강 리포트 생성 및 이메일 전송 함수 코드 >

```
# 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
```

- health_tips에 웹 서비스에서 출력할 모든 정보를 저장
- 함수의 return 값으로 health_tips를 지정하고 유저가 입력한 이메일로 전송

< 건강 사각형 시각화 함수 코드>

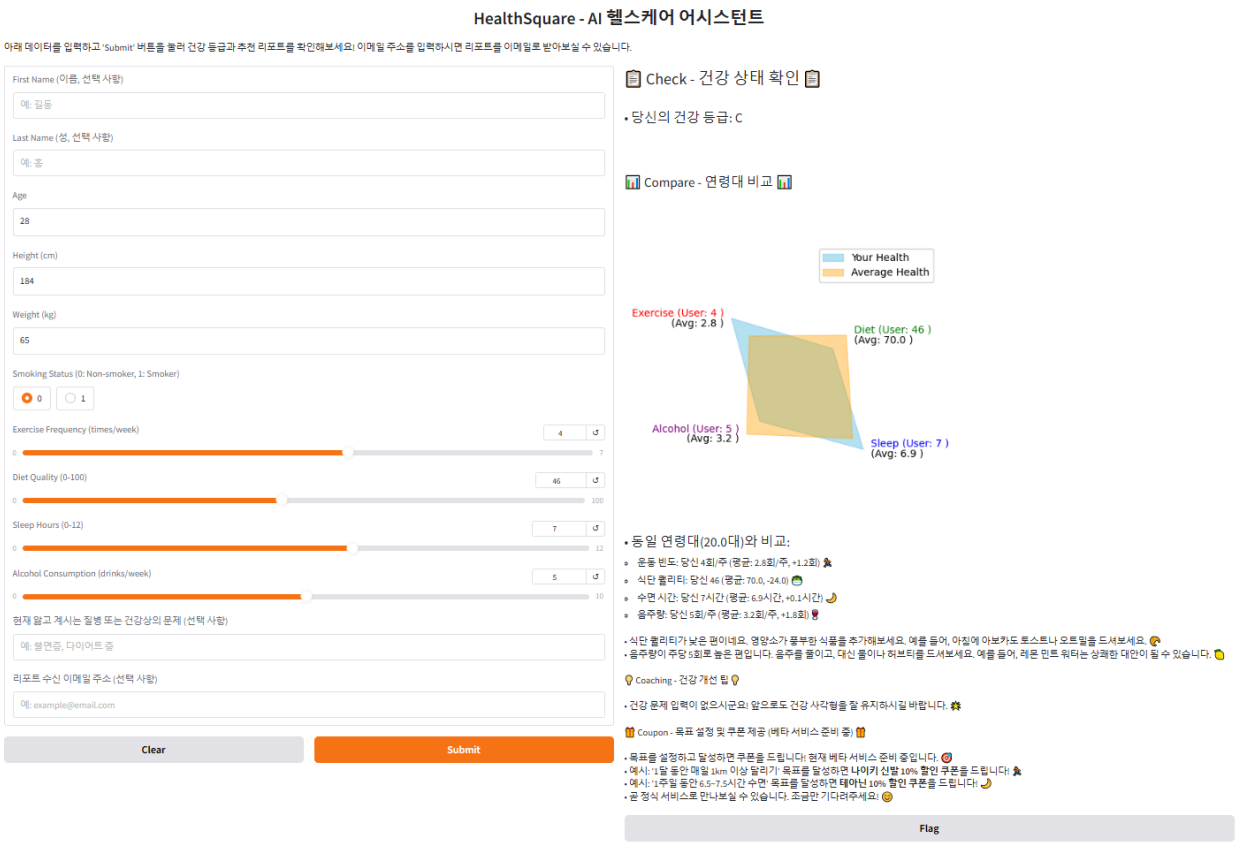
```python
# HealthSquare 시각화 함수 (Base64 인코딩 추가, units 정의)
def plot_health_square(user_values, avg_values, health_class, user_inputs, avg_inputs):
    center_x, center_y = 75, 75
    base_size = 30

    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)
```

```python
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
```

- 그래프의 기본 틀을 세팅하고 중앙을 기준으로 4개 지표(Exercise, Diet, Sleep, Alcohol)가 건강 사각형의 각 꼭짓점에 위치하도록 지정
- 모든 지표의 점수가 최저점일 때 매우 작은 정사각형, 최고점일 때 매우 큰 정사각형이 그려지도록 설정
- 유저 그래프와 연령대 평균 그래프를 하나의 화면에 출력시키되, 색깔을 다르게 하고 투명도를 조절하여 가시성을 개선
- 그래프의 각 꼭짓점이 어떤 지표를 의미하는지 표시하기 위해 라벨을 표기
- 유저 라벨과 평균 라벨이 겹쳐서 보이지 않도록 두 그래프의 꼭짓점 값 중에서 더 큰 값의 바깥쪽에 라벨이 표시되도록 설정
- 유저 라벨은 위쪽에, 평균 라벨은 아래쪽에 위치하도록 지정

6-3. 웹 서비스 시연

<Gradio 웹 서비스 화면 예시>



<이메일 리포트 화면 예시>

안녕하세요! HealthSquare에서 지준희님의 건강 리포트를 전달드립니다 ⚙️

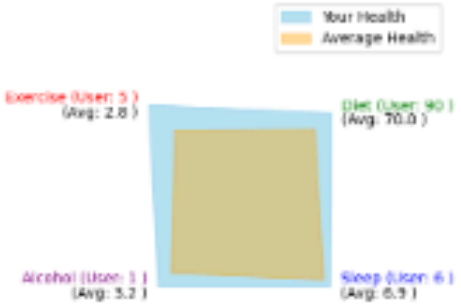
전반 등급 A

건강 유지 팁 리포트 📋

📋 Check - 건강 상태 확인 📋

장신의 건강 등급 A

📊 Compare - 연령대 비교 📊



동일 연령대(20대)와 비교:

- 운동 빈도: 장신 5회/주 (평균 2회/주, +2.8회) 🏃‍♂️
- 식단 밸런스: 장신 90 (평균 70.0, +20.0) 🍽️
- 수면 시간: 장신 6시간 (평균 6.9시간, -0.9시간) 🌙
- 주6일 장신 1회/주 (평균 3.2회/주, -2.2회) 🍺

- 식단 밸런스가 매우 좋으시네요! 현재 식단을 유지하시면서, 간식으로 견과류나 과일을 추가해보세요. 🍓
- 수면 시간이 짧은 편입니다. 최소 7~8시간 수면을 목표로 하세요. 취침 전 따뜻한 카모마일 차를 마시면 숙면에 도움이 됩니다. 🍵

💡 Coaching - 건강 개선 팁 💡

- 수험생 불면증, 스트레스로 인해 어려움을 겪고 계시군요. 몇 가지 구체적인 조언을 드릴게요. 💡
- 스트레스를 효과적 하는 규칙 있는 식단과 규칙적인 수면이 중요합니다. 💡
- 스트레스 완화에 도움이 되는 영양제로 **테아닌(L-theanine)**을 추천드립니다. 💊
- **테아닌**은 신경을 안정시키고 긴장을 완화하는 효과가 있어 스트레스 관리에 좋습니다. 녹차나 홍차에 포함되어 있지만, 영양제 형태로 섭취하는 것이 더 간편합니다. 🍵
- **테아닌**에 관심이 있으시다면 Now Foods의 L-Theanine 100mg을 추천드립니다. 약 1달 분량(60정)을 약 15,000원에 구매하실 수 있습니다. 🛒
- 또한, 스트레스를 완화하기 위해 가벼운 운동이 큰 도움이 됩니다. 날씨가 좋은 날 집 근처 공원에서 인터벌 러닝을 해보세요. 🏃‍♂️
- 인터벌 러닝은 3분 동안 목표에 달린 후 2분 동안 천천히 걷는 과정을 20~30분간 반복하는 방식입니다. 이렇게 하면 러닝에 집중하면서 스트레스를 해소하는 데 도움이 됩니다. ⏱️
- 수면의 질을 높이기 위해 규칙적인 수면 패턴을 유지하는 것이 중요합니다. 수면 개선에 도움이 되는 영양제로 **멜라토닌**을 추천드립니다. 💊
- **멜라토닌**은 수면 주기를 조절하는 데 도움을 주며, 불면증 완화에 효과적입니다. 특히, 코도, 멜니에 포함되어 있지만, 영양제로 섭취하면 더 간편합니다. 🍒
- **멜라토닌** 영양제로는 Nature's Melatonin 3mg을 추천드립니다. 약 1달 분량(60정)을 약 15,000원에 구매하실 수 있습니다. 🛒
- 수면 개선을 위해 취침 전 가벼운 스트레칭이나 명상도 추천드립니다. 다리 운동기 스트레칭&워밍 다리를 1분간 운동기을 해보세요. 긴장을 풀고 수면에 도움이 됩니다. 🌙

🎁 Coupon - 목표 설정 및 쿠폰 제공 (베타 서비스 준비 중) 🎁

- 수험생 목표를 설정하고 달성하면 쿠폰을 드립니다! 현재 베타 서비스 준비 중입니다. 🎯
- 예시: '1달 동안 매일 5km 이상 달리기' 목표를 달성하면 **나이키 신발 10% 할인 쿠폰**을 드립니다! 🏃‍♂️
- 예시: '1주일 동안 6.5~7.5시간 수면 목표를 달성하면 **테아닌 10% 할인 쿠폰**을 드립니다! 🌙
- 곧 정식 서비스로 만나보실 수 있습니다. 조금만 기다려주세요! 😊

더욱 향상된 HealthSquare 서비스를 확인해주세요! 🖼️

감사합니다.

HealthSquare 팀

7. 프로젝트 결과

7-1. 시사점

1) 알고리즘 다양성과 성능 최적화
LogisticRegression이 최종 모델로 선정되었지만 KNN, RandomForest, XGBoost 등 다른 알고리즘은 과대적합으로 인해 활용되지 못했다. 단일 모델에 의존하기보다 앙상블 기법 등을 도입해 여러 모델의 장점을 결합하는 시도를 취했다. 또한 하이퍼파라미터 튜닝도 Grid Search CV 외에도 Randomized Search CV와 Bayesian Optimization을 추가적으로 적용해서 효율성을 높였다. 최적화된 모델에서는 과대적합 및 과소적합 문제가 상당 부분 해소됐지만 데이터셋 확장 시 재발 가능성이 있으므로 교차 검증을 강화할 필요가 있다.

2) 데이터셋 왜곡 및 편향
: 기존 데이터셋의 Target(Health_Score) 값이 왜곡되거나 편향되어 있었기 때문에 데이터를 전처리하고 특성 공학을 활용하여 모델을 학습시켜도 이상적인 수준까지 성능이 개선되지 않았다. 이는 데이터셋 자체의 객관성과 다양성이 부족했음을 시사하며 머신러닝 모델의 성능이 데이터셋의 품질에 얼마나 크게 좌우될 수 있는지를 보여준다. 해당 프로젝트에서는 기존 Target 값을 대체하는 신규 Target(New_Health_Score)을 생성하여 합리적인 수준에서 유저가 받아들일 수 있는 웹 서비스를 개발할 수 있었다.

7-2. 개선점

1) 웹 서비스를 통한 데이터 수집
: 데이터분석과 머신러닝을 통해 최적화된 모델을 기반으로 웹 서비스를 구현했다. 기존의 데이터셋만으로는 유의미한 인사이트를 도출하는 모델을 구축하기가 어렵기 때문에 웹 서비스를 통해 입력받은 유저의 정보를 수집하여 데이터 샘플을 늘리면 지속적으로 머신러닝 모델을 개선할 수 있을 것으로 판단된다. 웹 서비스를 통해 데이터를 수집하는 과정에서 이상치로 판단되는 값은 수시로 제거하거나 변환하여 모델의 성능이 저하되지 않도록 해야 할 것이다. 또한 오디오나 비디오 형태의 파일도 추가하여 유저가 보다 역동적인 환경에서 서비스를 이용할 수 있도록 개선할 계획이다.

2) 사용자 참여 유도 전략 필요성
: 웹 서비스에서 건강등급(Health_Class)과 함께 개선 팁을 담은 이메일 리포트를 통해 개인별로 맞춤화된 조언을 전달한 것은 사용자 참여를 끌어내는 데 핵심적인 요소였다. 단순히 점수를 보여주는 데에만 그치지 않고 건강 사각형을 통해 동일 연령대와 비교할 수 있도록 하고 건강점수를 개선할 수 있는 구체적인 방법을 제시함으로써 동기부여를 심어줄 수 있었다. 이는 헬스케어 서비스가 정보를 제공하는 것을 넘어 사용자의 행동을

유도해야만 지속가능한 서비스로 남을 수 있음을 보여준다. 결과적으로 사용자 참여를 유도하는 설계는 기술적 완성도만큼이나 서비스 성공에 중요한 요소임을 알게 되었다.