#PLAYDATA MINIPROJECT2

# 심장질환 AI 예측 서비스

TEAM 우병목

# 목차

# 1. 프로젝트 개요

- 프로젝트 소개

## Personal Key Indicators of Heart Disease

2020 annual CDC survey data of 400k adults related to their health status

Data Card | Code (156) | Discussion (16)

### About Dataset

**Usability** ⓘ
10.00

**License**
CC0: Public Domain

**Expected update frequency**
Annually

### Key Indicators of Heart Disease

**2020 annual CDC survey data of 400k adults related to their health status**

CDC에서 2020년간 40만명의 성인을 대상으로 전화설문을 통해 심장질환 유무를 수집한 데이터(캐글)

**머신러닝 모델을 통한 심장질환 유무 예측**

▼

**모델 활용 심장질환 예측 웹 구현**

## 1. 프로젝트 개요

- 구성원 역할 및 소개

### 우상욱

**역할**

- EDA
- 전처리
- 머신러닝 모델링
- 웹 구현

### 민병창

**역할**

- 전처리
- 머신러닝 학습 및 선택
- 머신러닝 모델링
- 시각화

### 김경목

역할

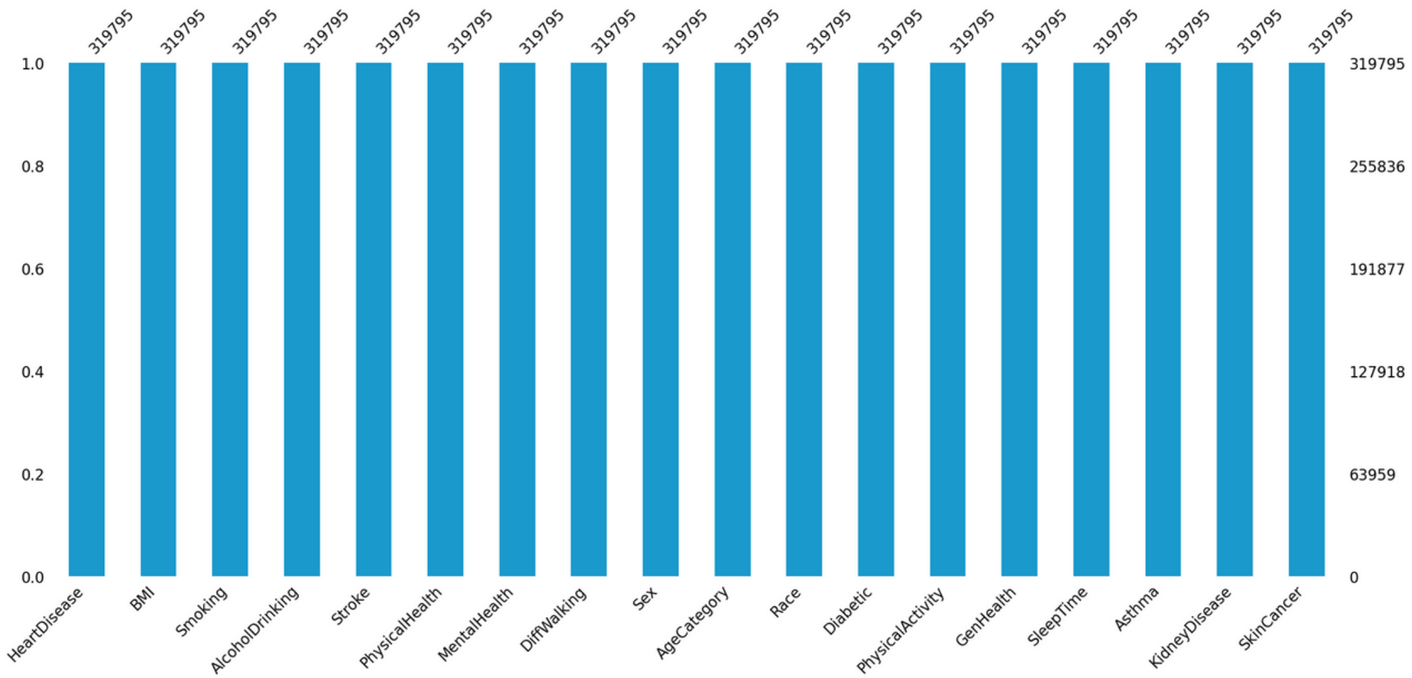
- 시각화
- 머신러닝 학습
- 머신러닝 모델링

## 1. 프로젝트 개요

- 개발 환경 및 활용 라이브러리
  - 개발환경 : Python 3.9.13
  - 라이브러리
    - 데이터 수집 및 전처리 : pandas, numpy
    - 시각화 : matplotlib , Seaborn
    - 머신러닝 : scikit-learn, lightgbm , xgboost ,catboost
    - 웹구현 : streamlit
  - Devops
    - Git, Notion, Canva

# 2. EDA & 전처리

- 중복값 처리 / 결측치 확인




msno 라이브러리 활용 결측치 시각화 -> 결측치 없음

df.drop_duplicates() -> 중복행 18078개 삭제

## 2. EDA & 전처리

- 변수 형태별 분리

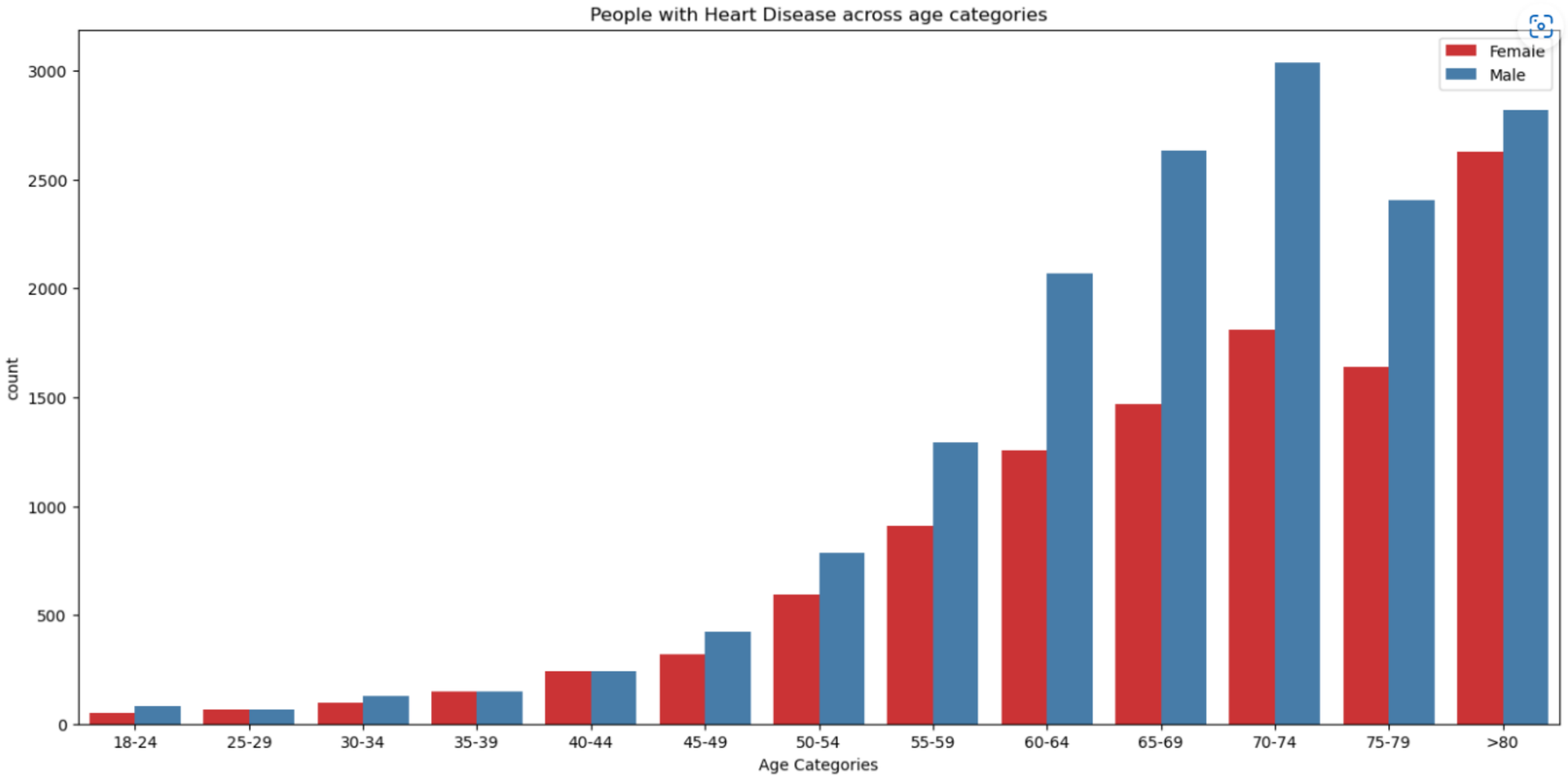
```
BMI                float64
SleepTime          float64
PhysicalHealth     float64
MentalHealth       float64
HeartDisease        object
Asthma              object
GenHealth           object
PhysicalActivity    object
Diabetic            object
Race                object
Sex                 object
KidneyDisease       object
DiffWalking         object
Stroke              object
AlcoholDrinking     object
Smoking             object
AgeCategory         object
SkinCancer          object
```

- 총 18개의 컬럼, 319,795 행 개수
- 수치형 컬럼 : 4개
- 범주형 변수 : 14개(타겟 변수 1개 포함)

## 2. EDA & 전처리

- 범주형 변수 시각화

### 심장질환을 가진 사람의 성별, 나이 분포 시각화

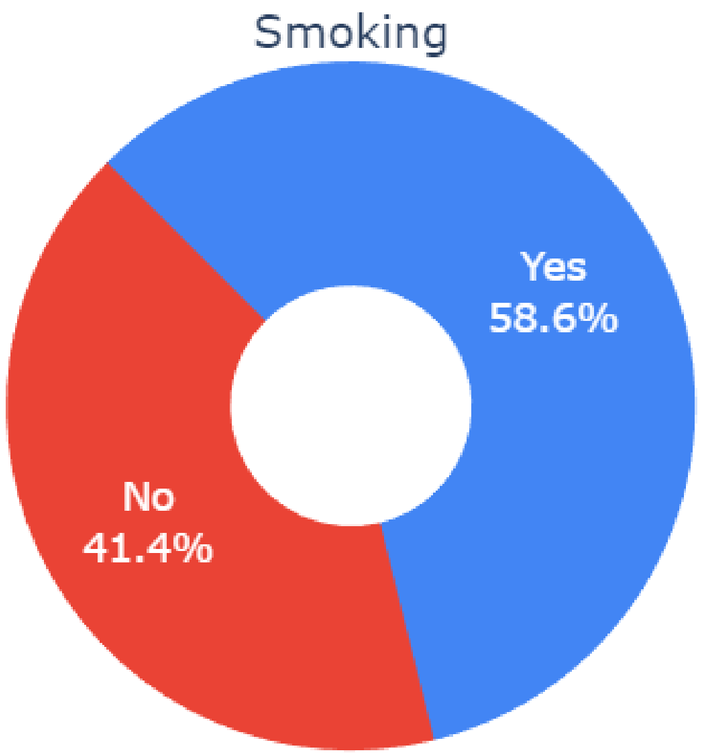



심장질환을 가진 사람은 나이가 많을 수록, 남자일수록 더 많이 분포 된 것으로 보인다.

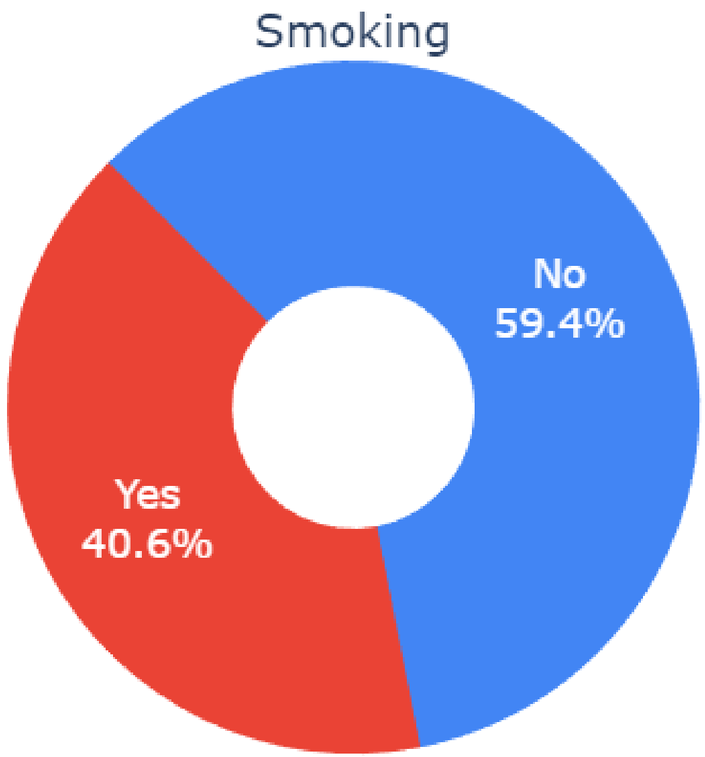
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

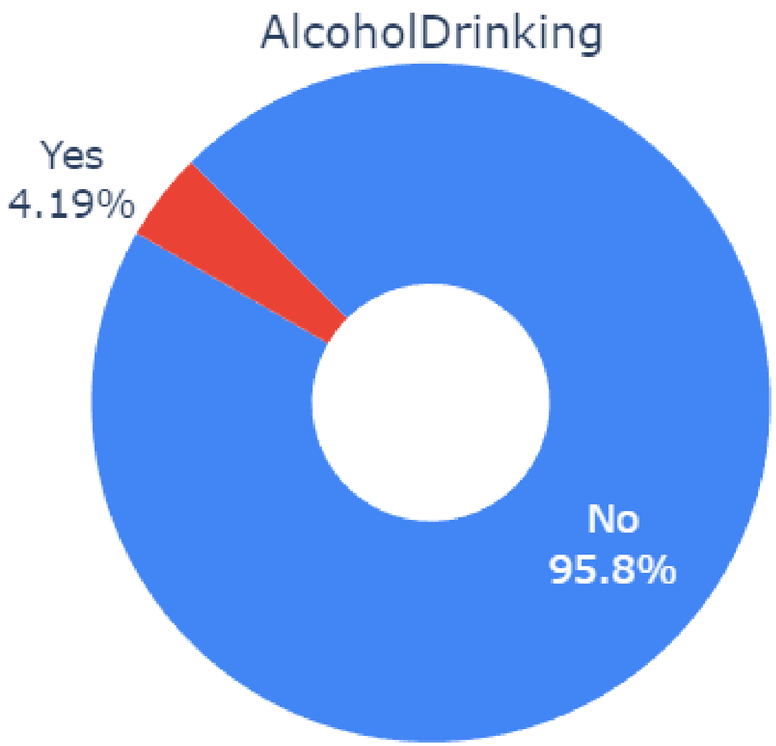



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 흡연율이 높다.

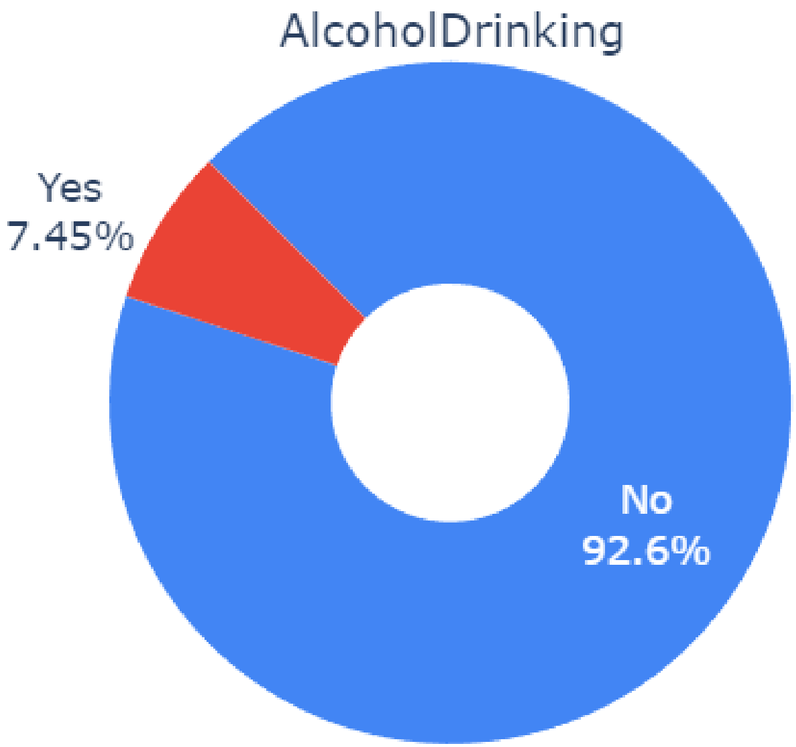
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

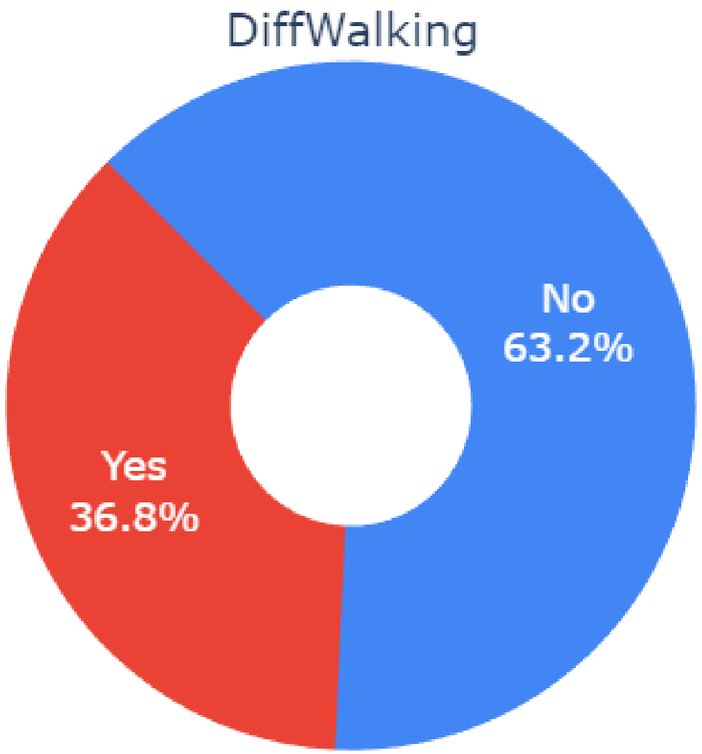



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 과음자 비율이 적다

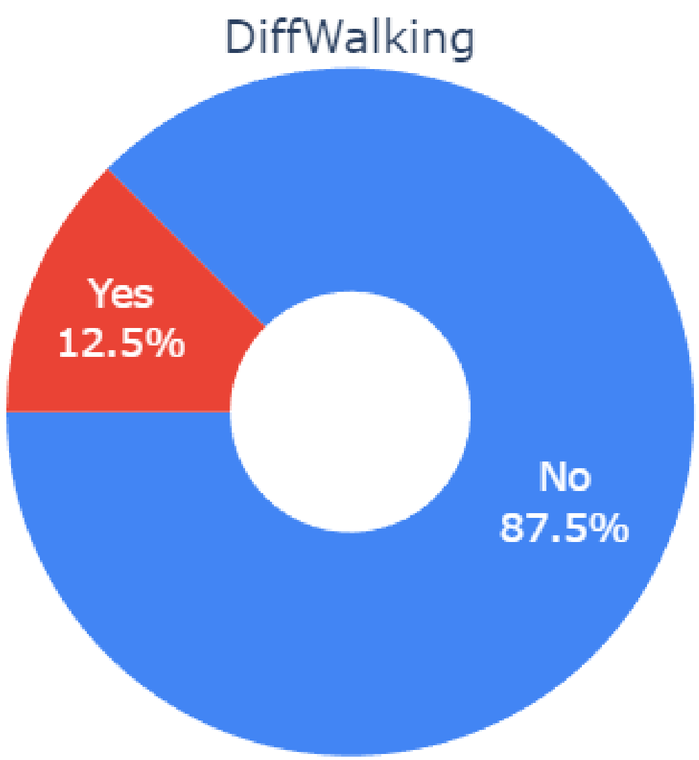
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

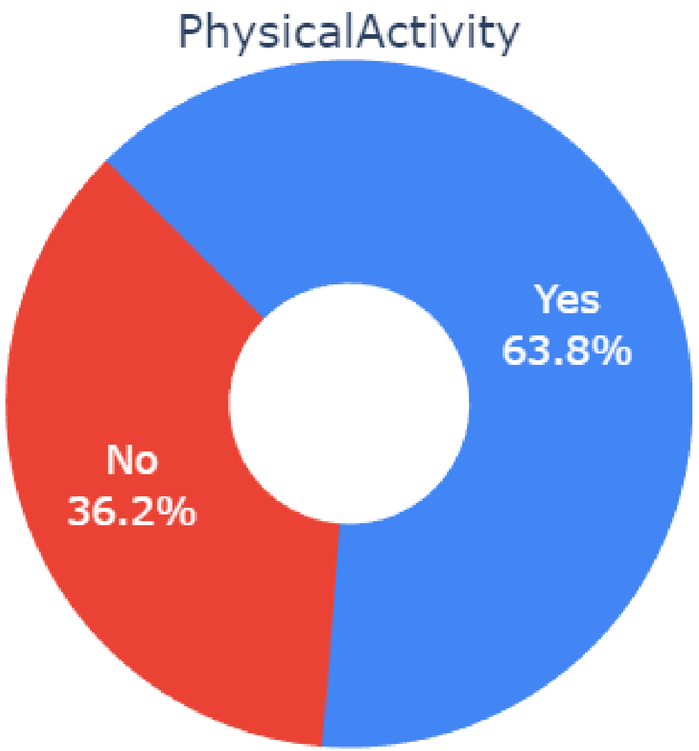



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 계단을 오를 때 어려움을 느끼는 비율이 높다

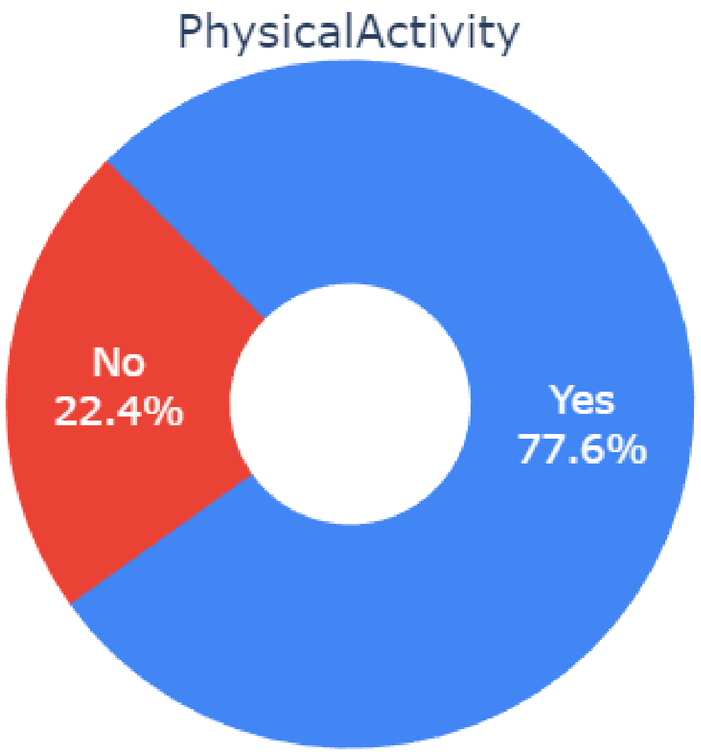
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

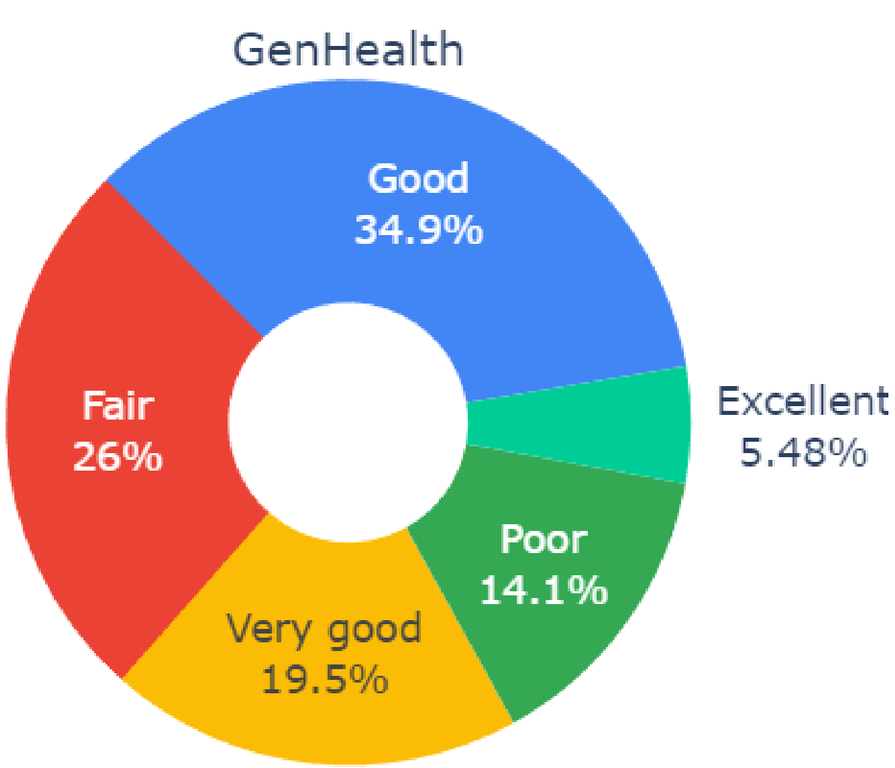



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 운동을 하지 않는 비율이 높다

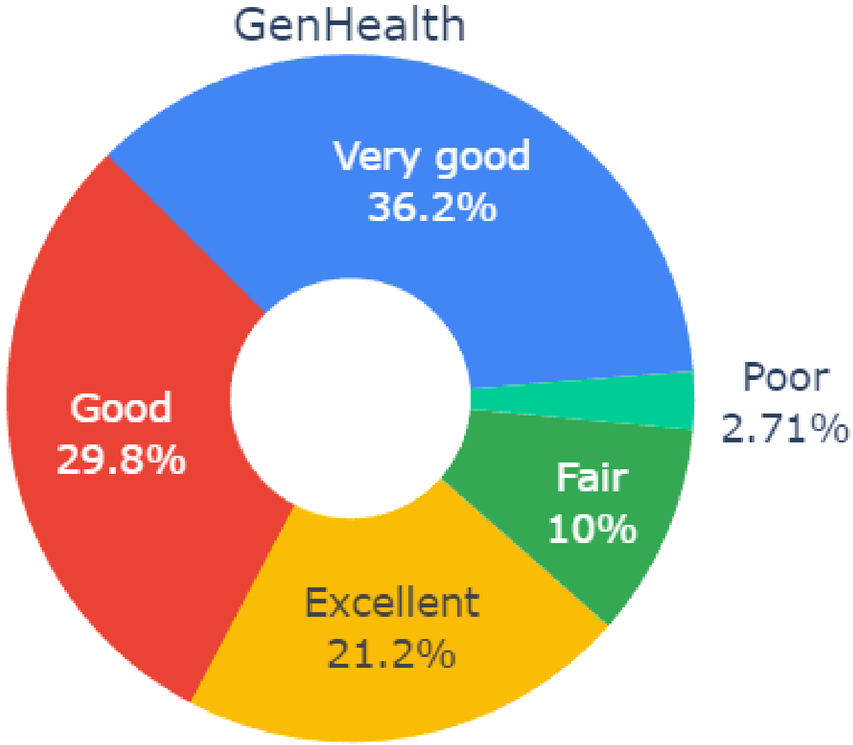
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

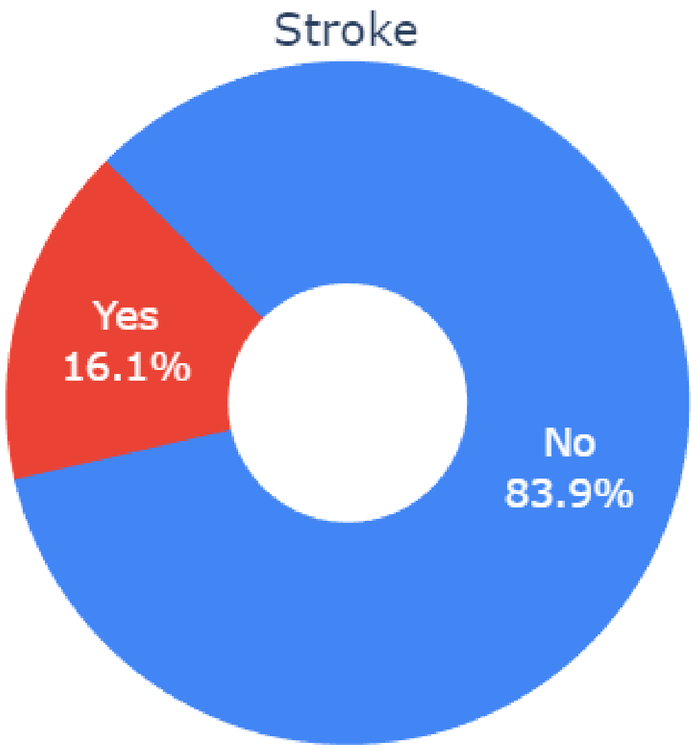



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 자신이 건강하지 않다 생각하는 비율이 높다

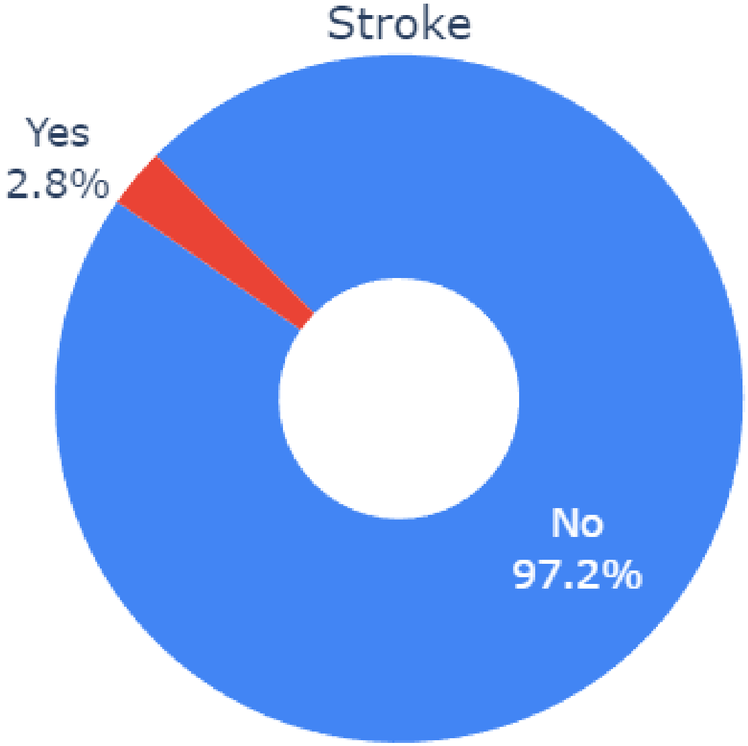
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

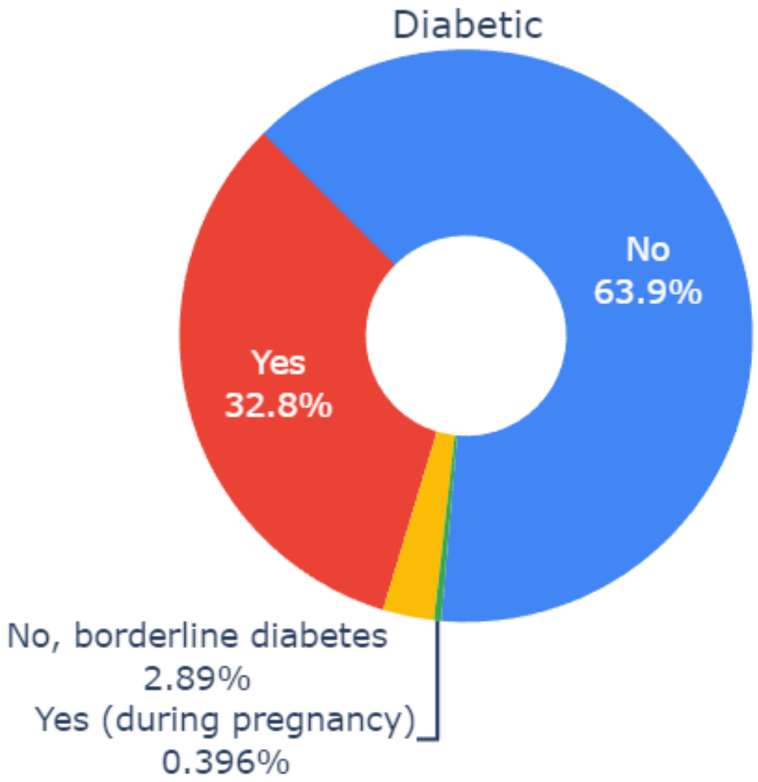



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 뇌졸중 경험 비율이 높다

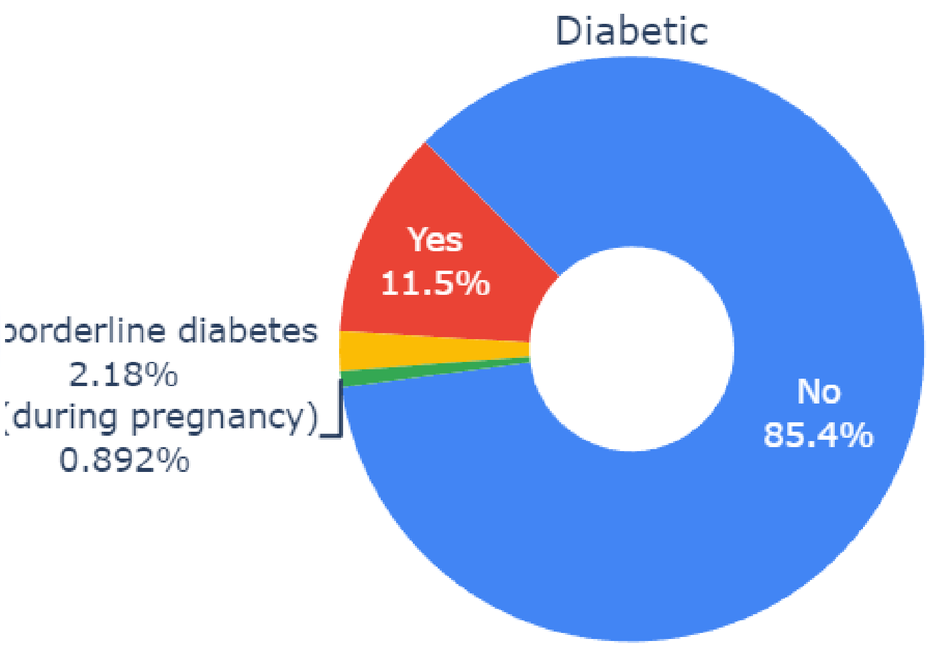
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

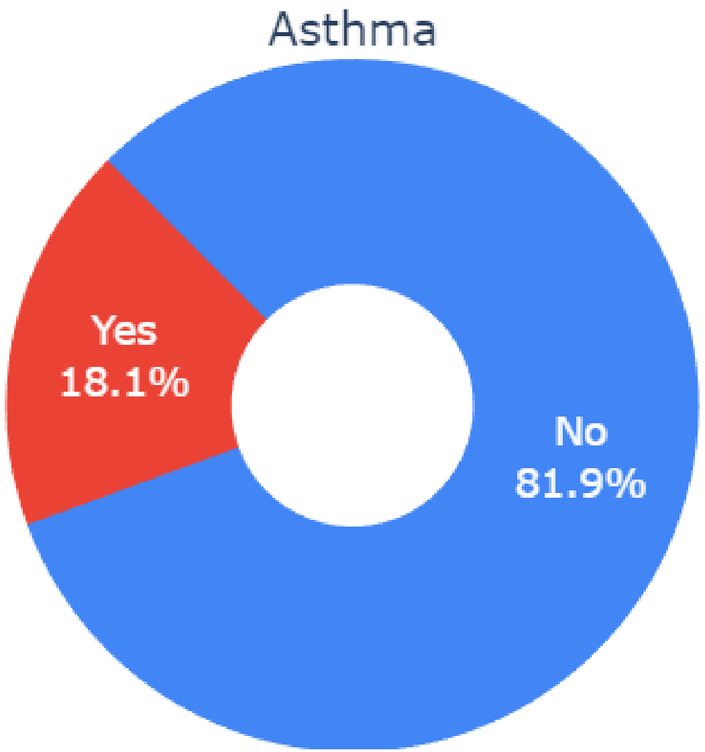



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 당뇨병 경험 비율이 높다

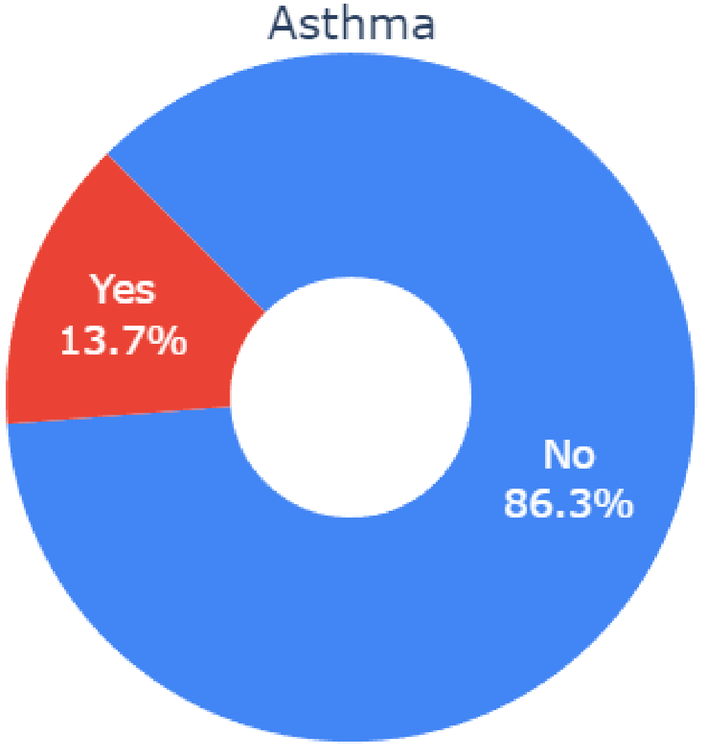
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

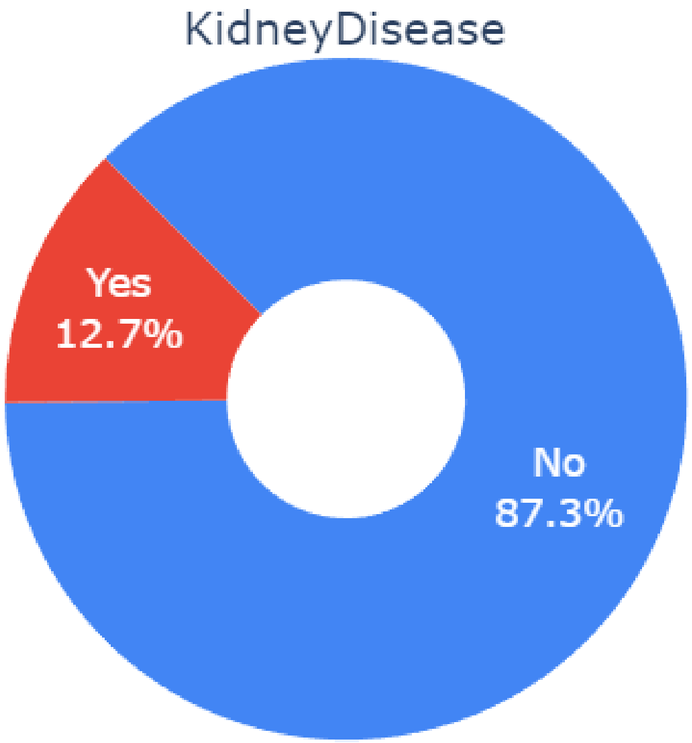



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 천식 경험 비율이 높다

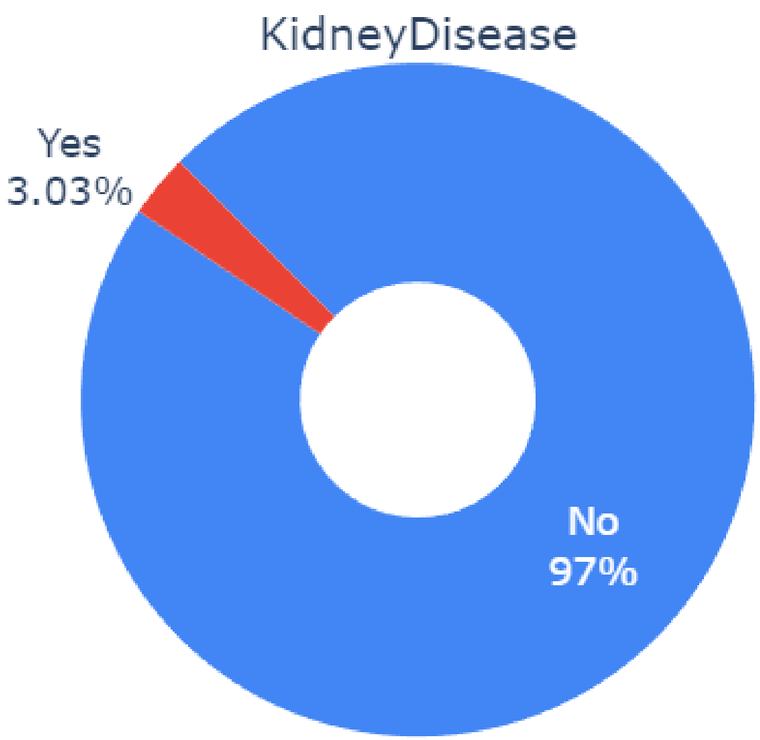
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹

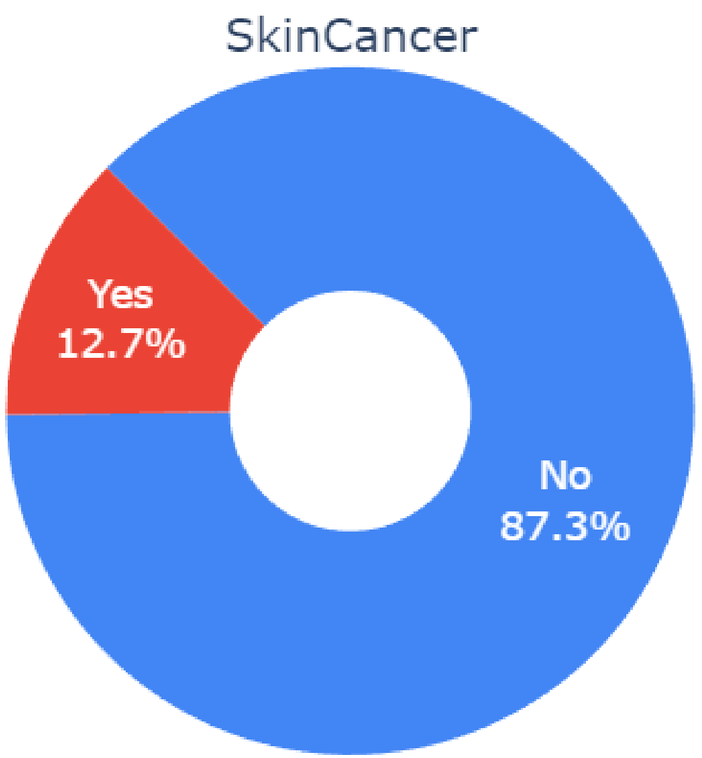



심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 신장질환 경험 비율이 높다

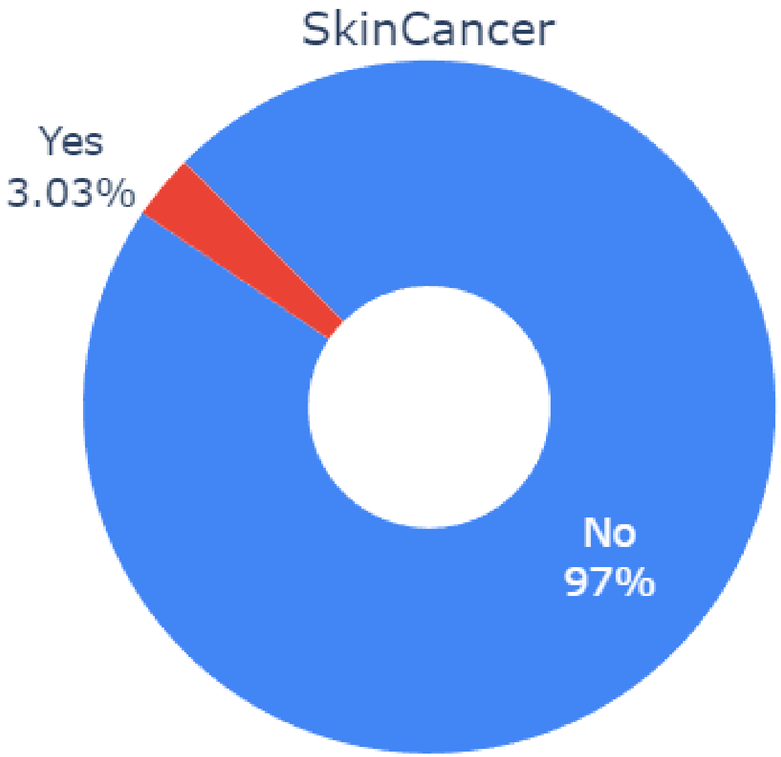
## 2. EDA & 전처리

- 심장질환과의 상관관계

심장질환이 있는 그룹




심장질환이 없는 그룹




심장질환을 가진 그룹은 심장질환을 가지지 않은 그룹에 비해 피부암 경험 비율이 높다

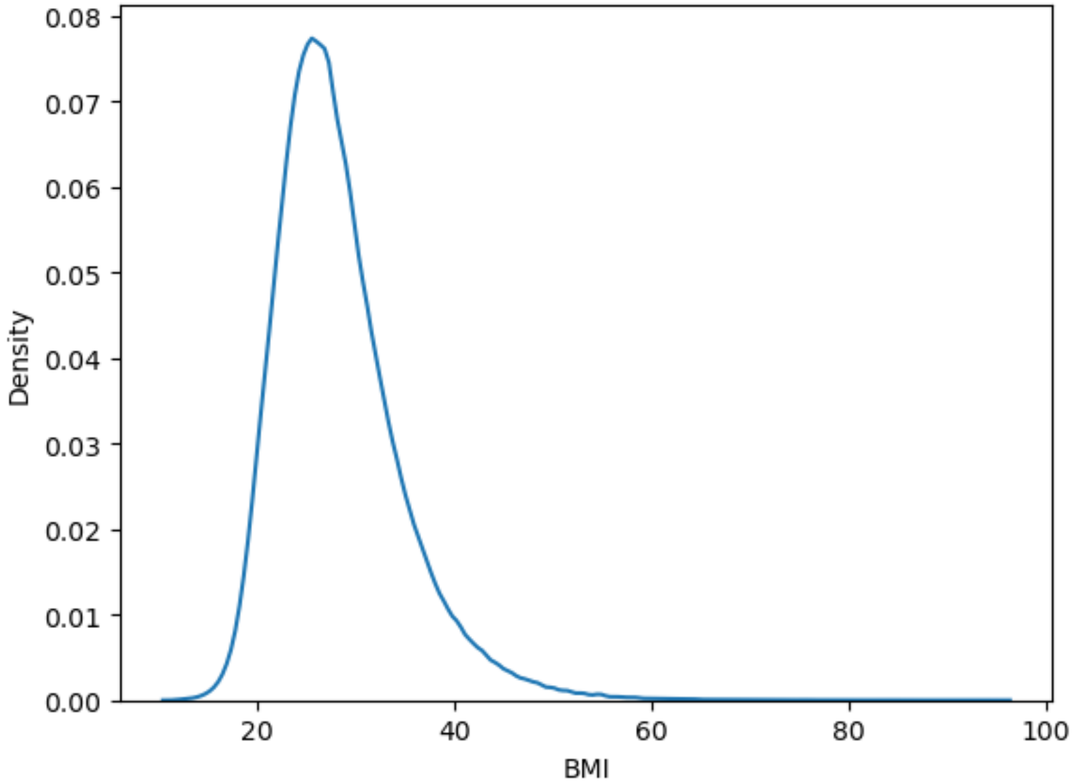
## 2. EDA & 전처리

- 수치형 변수 시각화


수치형 변수 간 다중공선성은 없다고 판단

## 2. EDA & 전처리

- 연속형 변수 시각화

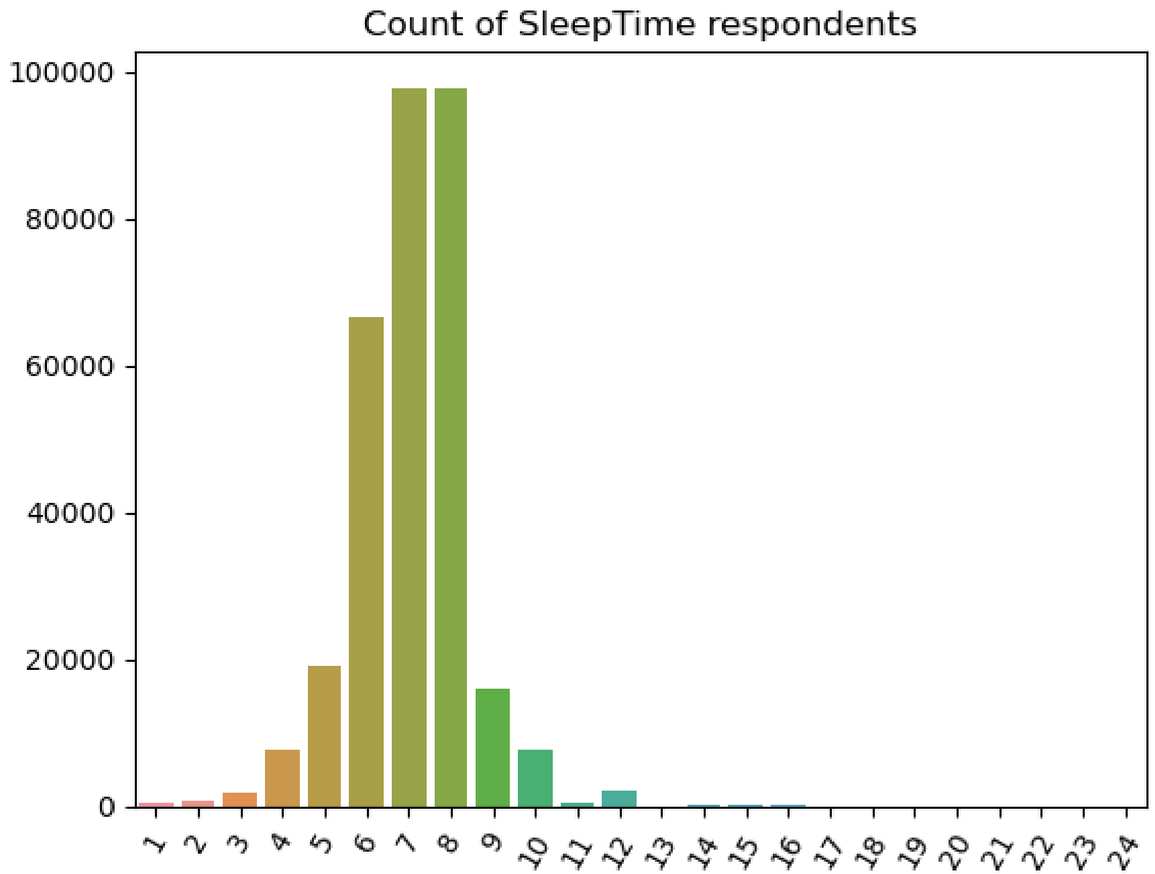



BMI 분포 시각화

| | 심장질환 o | 심장질환 x |
|---|---|---|
| BMI 평균값 | 29.4 | 28.2 |

## 2. EDA & 전처리

- 연속형 변수 시각화

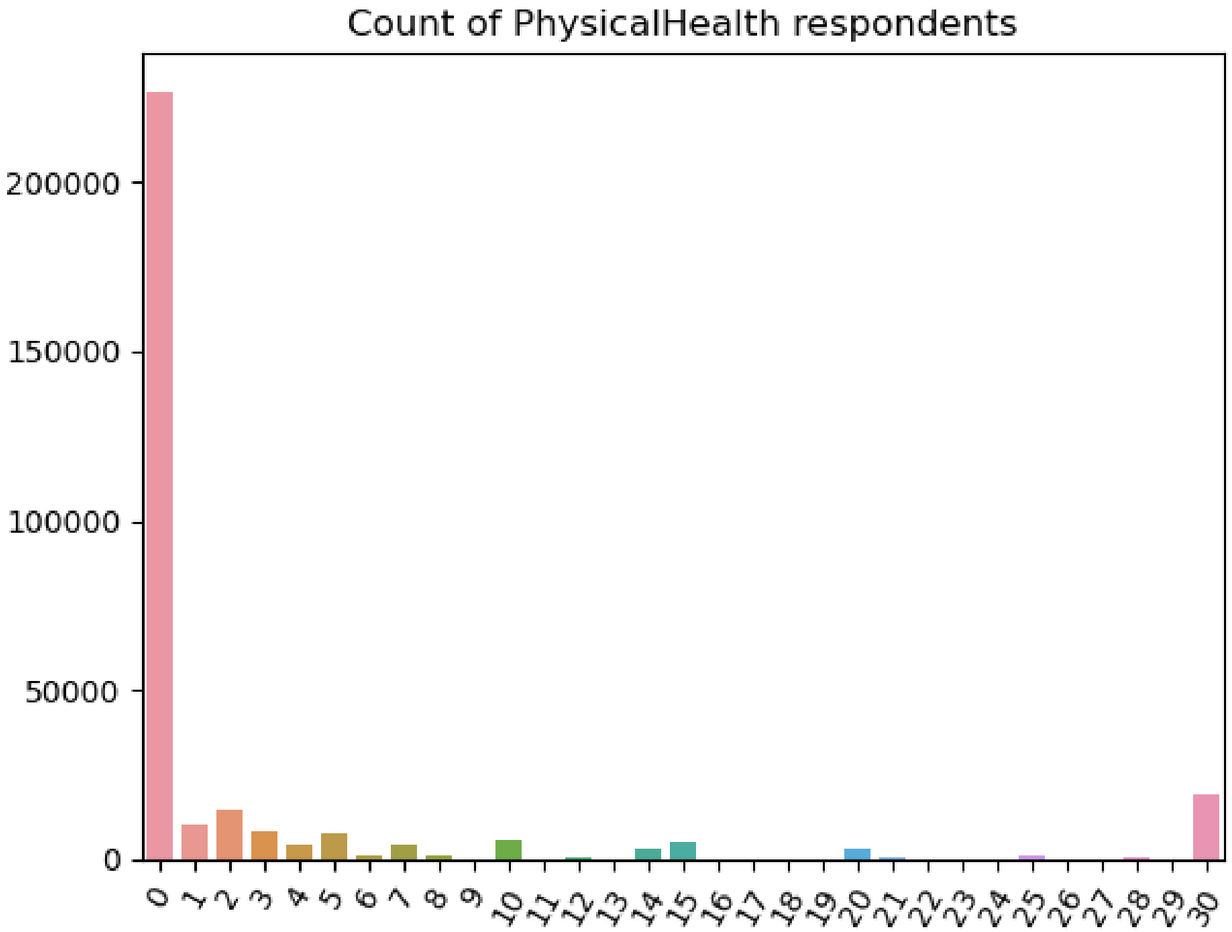



하루 평균 몇 시간씩 자는지에 대한 응답수

| | 심장질환 o | 심장질환 x |
|---|---|---|
| SleepTime 평균값 | 7.13 | 7.09 |

## 2. EDA & 전처리

- 연속형 변수 시각화

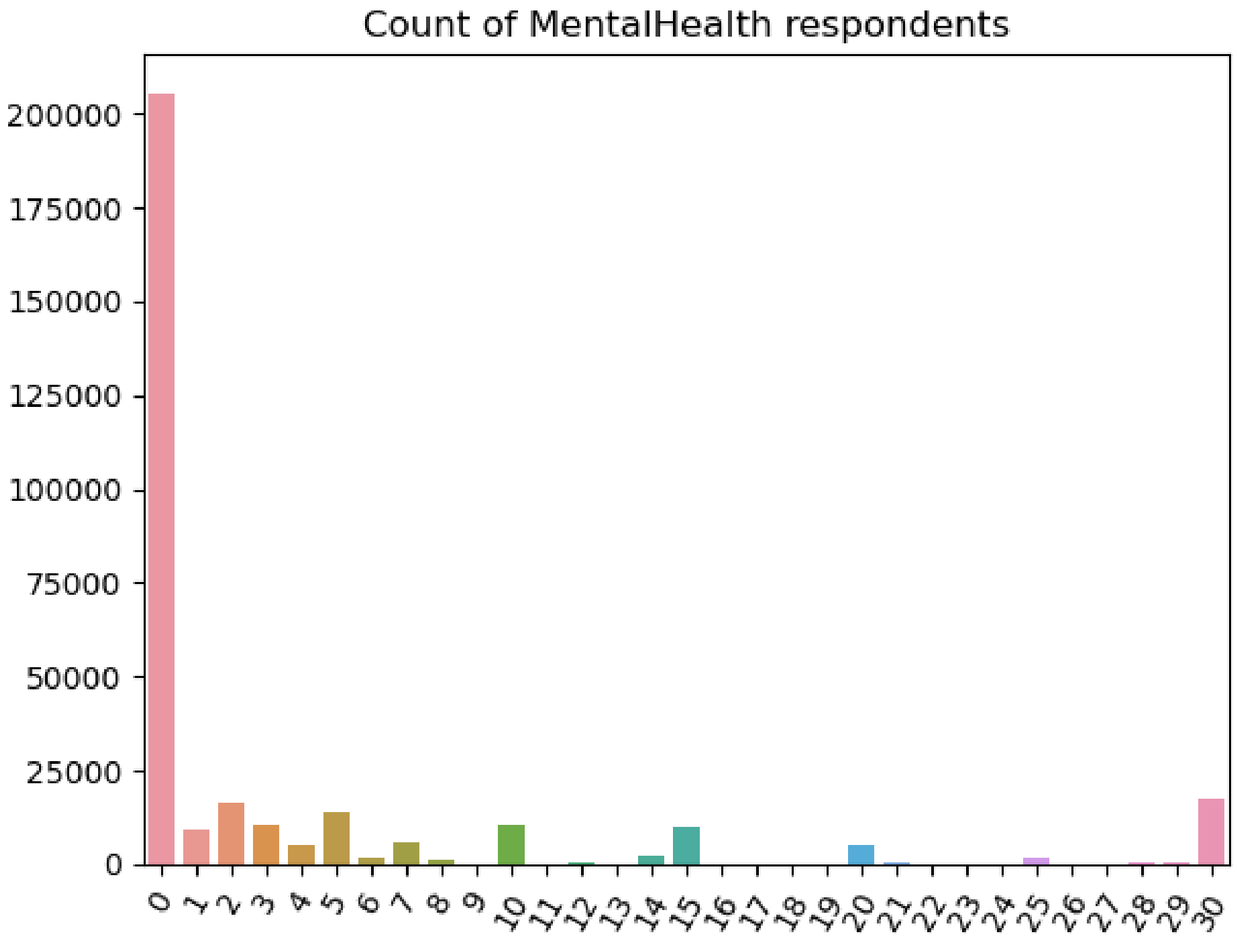



최근 30일간 신체가 건강하지 못했던 일 수

| | 심장질환 o | 심장질환 x |
|---|---|---|
| PhysicalHealth 평균값 | 7.80 | 2.95 |

## 2. EDA & 전처리

- 연속형 변수 시각화

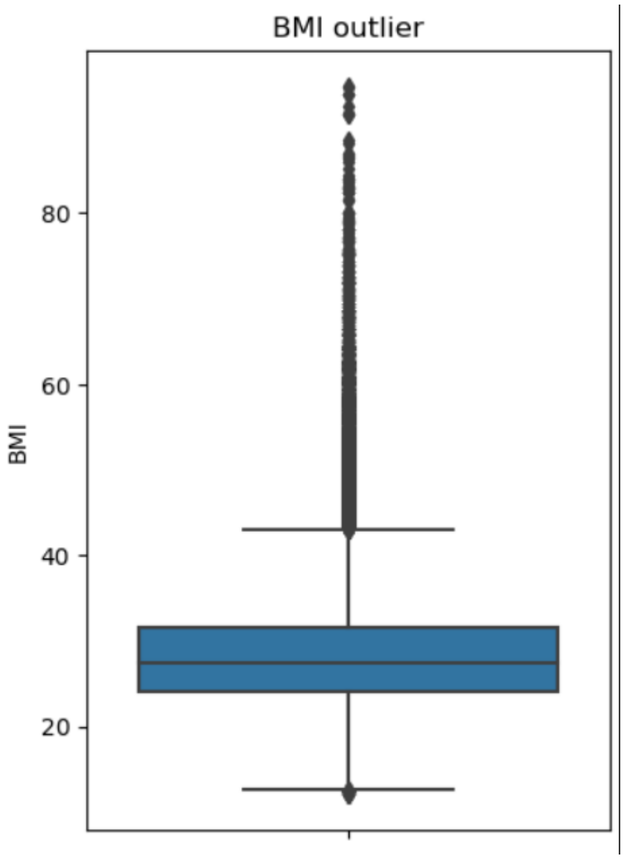


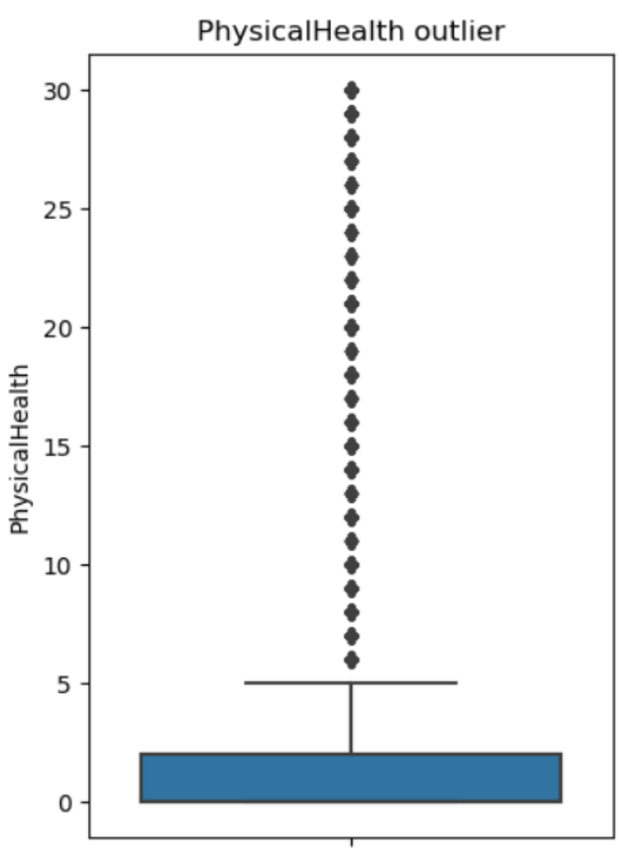

최근 30일간 정신이 건강하지 못했던 일 수

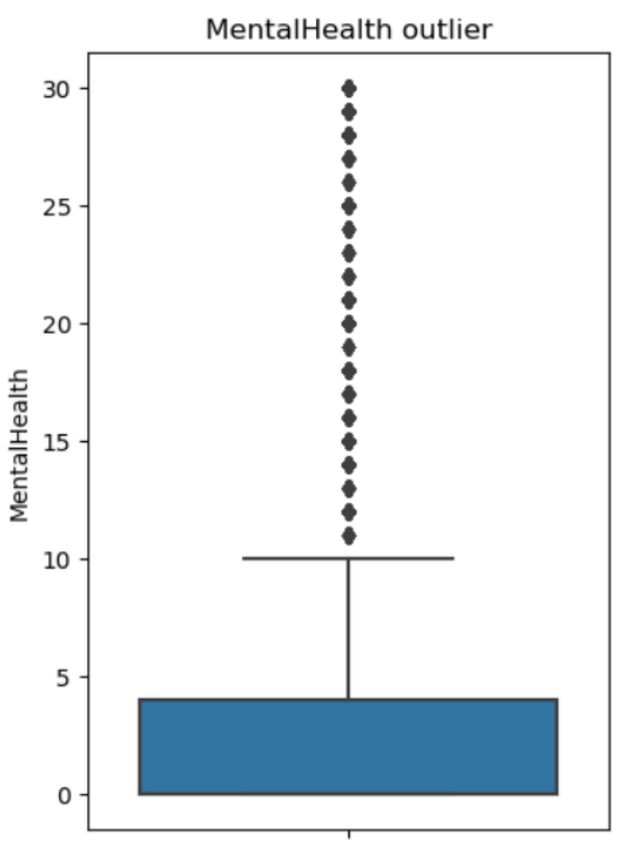
| | 심장질환 o | 심장질환 x |
|---|---|---|
| MentalHealth 평균값 | 4.64 | 3.82 |

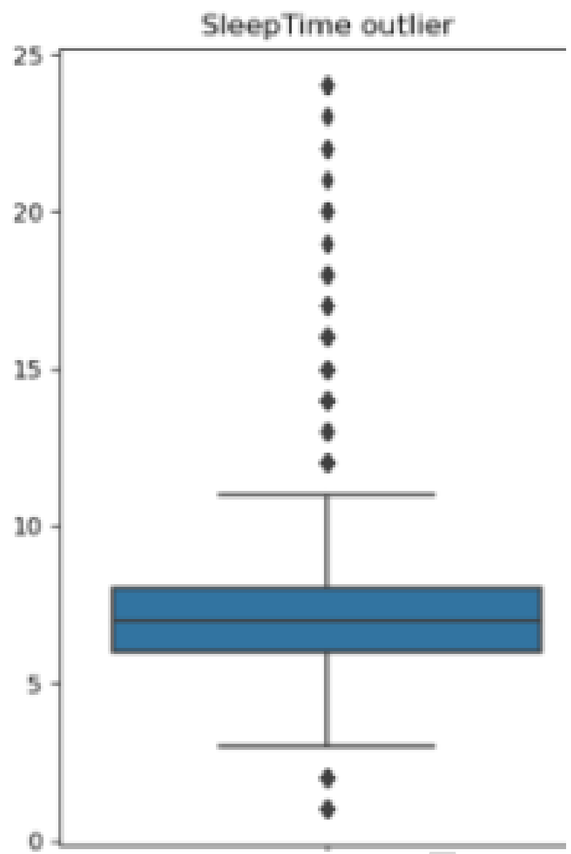
## 2. EDA & 전처리

- 이상치 분석



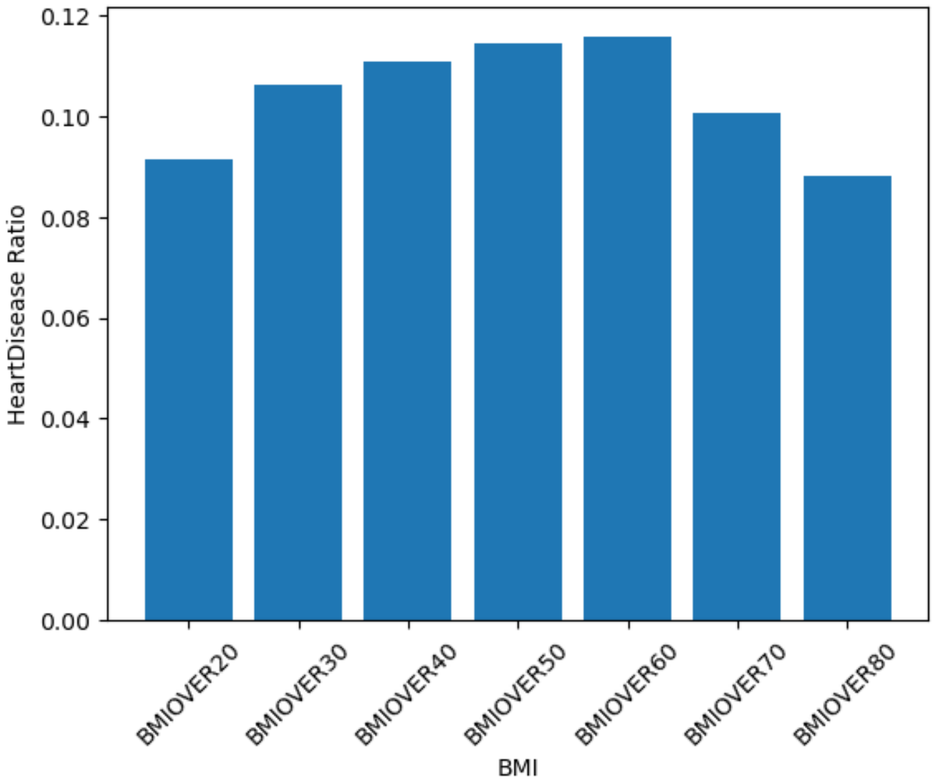



## 2. EDA & 전처리

- 이상치 분석




따라서 BMI 지수가 특정 지수를 넘을 때
심질환자 비율을 나타낸 그래프

심지어 BMI가 70을 넘었을 때부터
심질환자 비율이 많이 하락하는 것을 볼 수 있음

▼

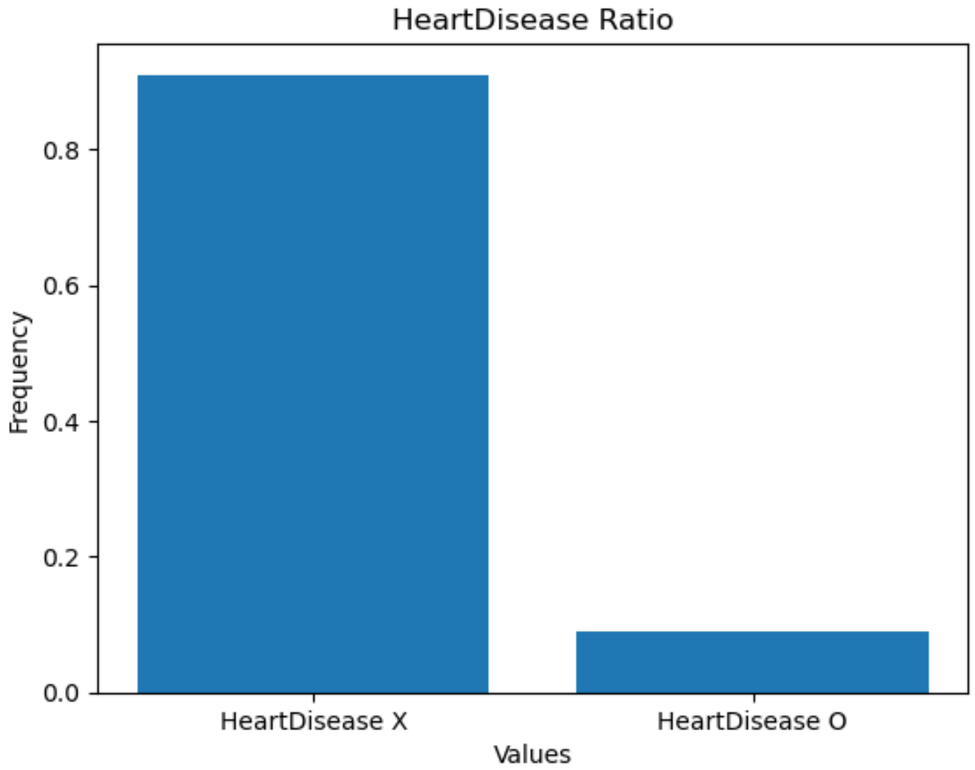
BMI 70 이상은 이상치로 판단하고 삭제 시도

▼

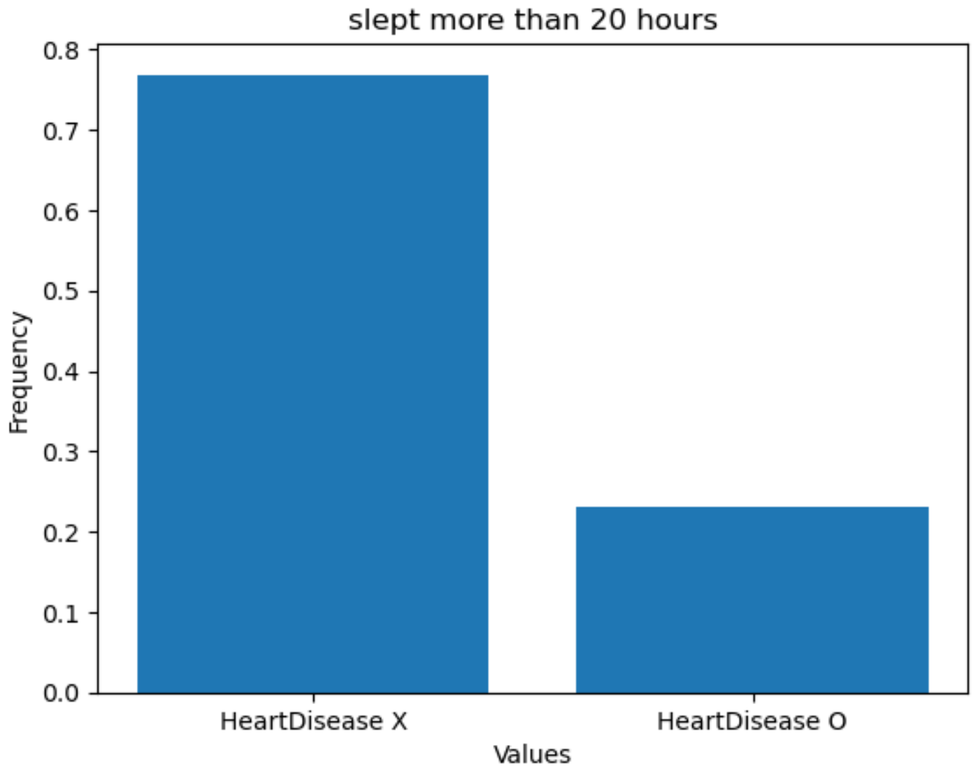
**예측 성능 하락(이상치 삭제 X)**

## 2. EDA & 전처리

- 이상치 분석




모든 사람의 심질환 비율




수면 시간이 20시간 이상인 사람들의 심질환 비율

보수적으로 수면시간이 24시간인 경우만 이상치로 판단 후 삭제 시도

▼

예측 성능 하락(이상치 삭제 X)

## 2. EDA & 전처리

- 인코딩

범주형 변수 中 유니크 값 2개

▼

라벨 인코딩

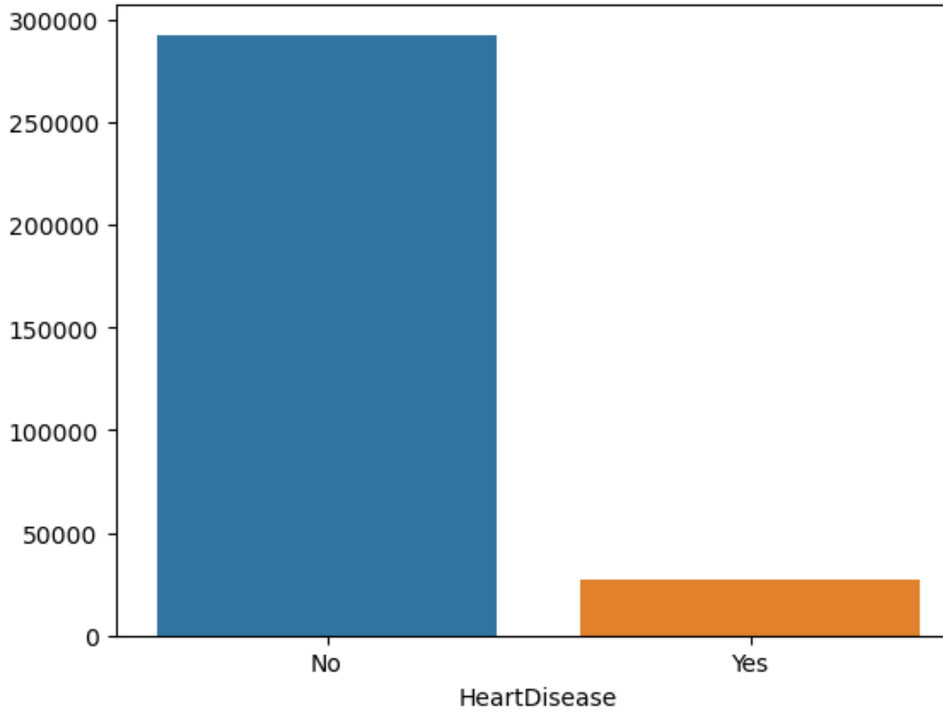
범주형 변수 中 유니크 값 3개 이상

▼

원핫 인코딩

# 3. 모델링

- 분류 모델별 정확도 비교




타겟 변수 불균형 -> SMOTE, SMOTEOMEK

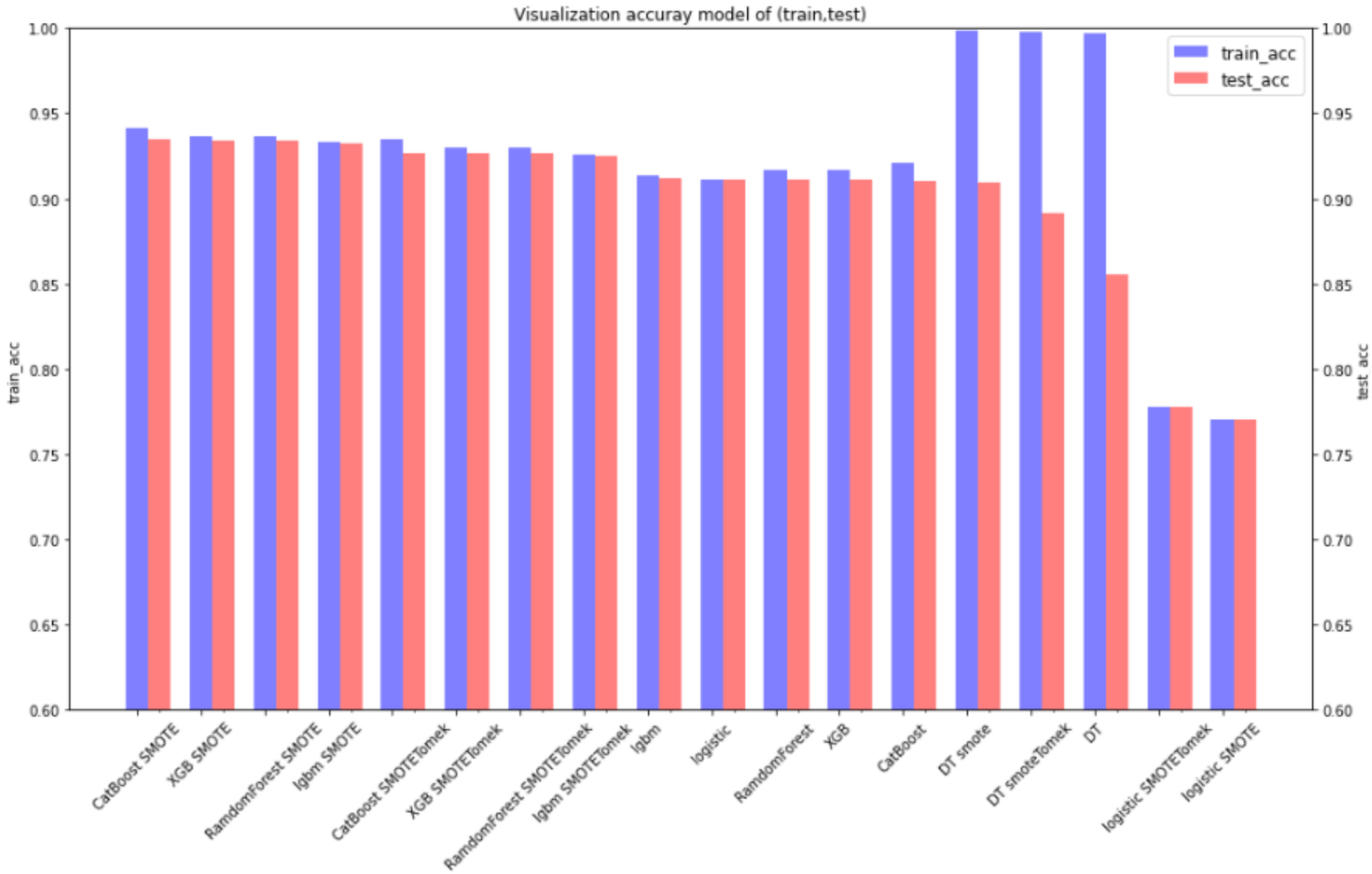
```python
df = pd.read_csv('./../0.data/heart_2020_final.csv')

X = df.iloc[:,1:].values
y = df['HeartDisease']

X_smote,y_smote = SMOTE(random_state = 42).fit_resample(X,y)

X_train_ns,y_train_ns = SMOTETomek(sampling_strategy=0.5).fit_resample(X,y)
```

기존 데이터 , SMOTE 데이터 , SMOTETomek 데이터로 모델별 학습

# 3. 모델링

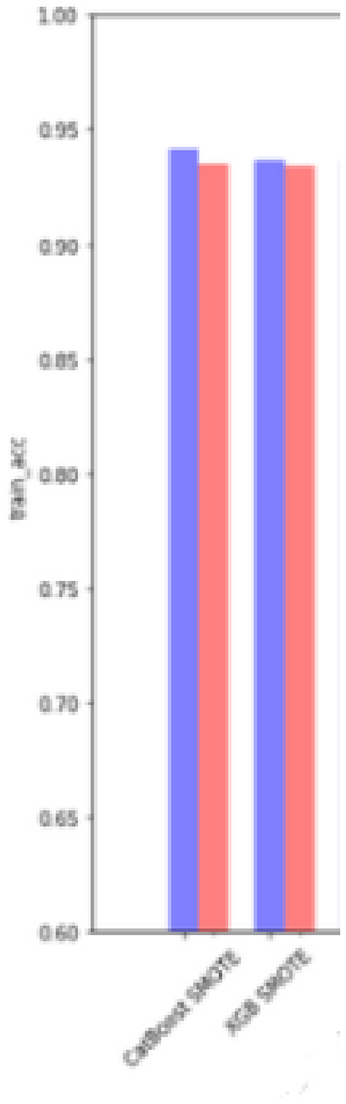
- 분류 모델별 정확도(accuracy) 비교

## 파라미터 default 값으로 예측 / K_FOLD = 5 평균 accuracy 값



## 3. 모델링

- 상위모델 선정




accuracy 지표, 과적합 여부 판단 후 상위 2개 모델 선정

▼

CATBOOST, XGBOOST

## 3. 모델링

- 하이퍼 파라미터 튜닝(그리드 서치)

### CATBOOST

bagging_temperature = 0, depth = 9,
l2_leaf_reg = 3,
learning_rate =0.1

| | train | test |
|---|---|---|
| 정확도 | 0.9336 -> 0.9532 | 0.9325 -> 0.9357 |
| 재현율 | 0.8933 -> 0.9240 | 0.8923 -> 0.9009 |

심질환자가 맞는데, 아니라고 예측했을 때 모델의 치명적 단점
-> 중요지표로 선정

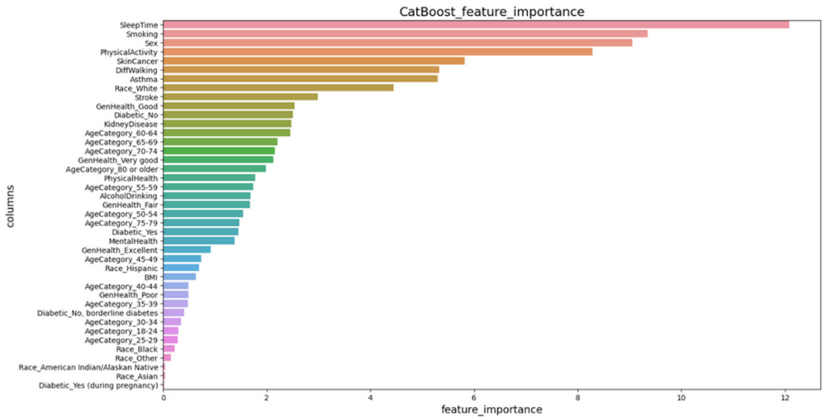
### XGBOOST

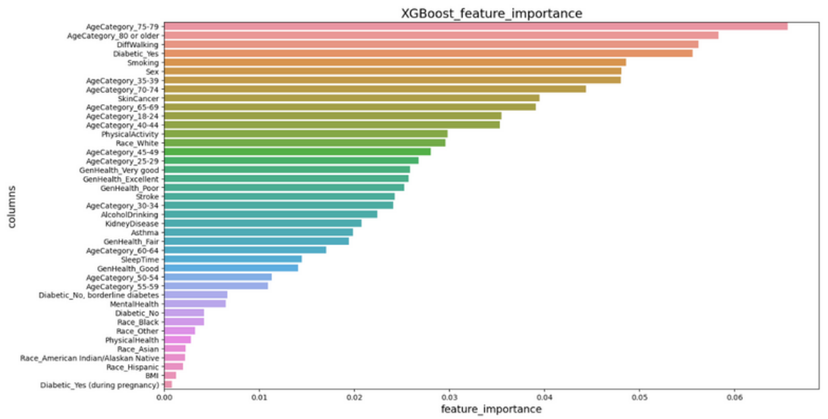
colsample_bytree = 0.8,
learning_rate = 0.1,
max_depth = 5,
min_child_weight = 4,
n_estimators = 200,
subsample = 0.9

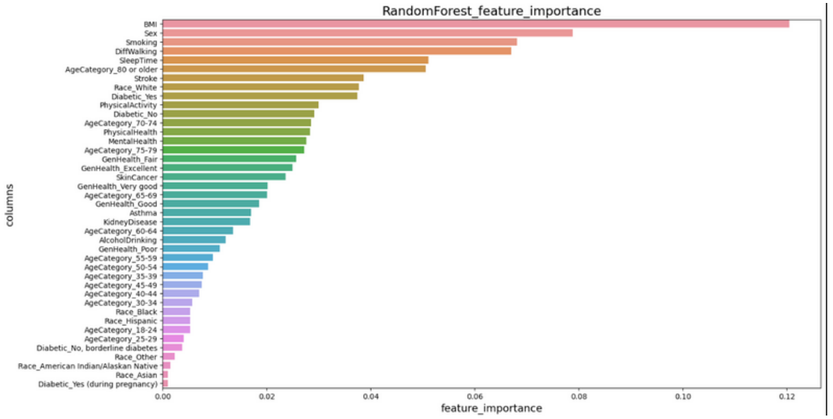
| | train | test |
|---|---|---|
| 정확도 | 0.9366 -> 0.9562 | 0.9342-> 0.9435 |
| 재현율 | 0.8970 -> 0.9195 | 0.8946 -> 0.9065 |

# 3. 모델링

- 특성 선택




CatBoost, XGboost, RandomForest에서 추출한 특성 중요도를 바탕으로
특성 중요도 하위 독립 변수 4개 선정

▼

Race, GenHelath, Diabetic, MentalHealth

## 3. 모델링

- 특성 선택

평가 점수 산정을 위해 튜닝 된 CatBoost만 사용해서
특성 하나씩 Drop 하며 비교

| Drop_columns | train 정확도 | test 정확도 | train 재현율 | test 재현율 |
|---|---|---|---|---|
| None | 0.9532 | 0.9357 | 0.9240 | 0.9009 |
| Race | 0.9503 | 0.9329 | 0.9197 | 0.8976 |
| GenHeatlh | 0.9390 | 0.9228 | 0.9123 | 0.8916 |
| Diabetic | 0.9486 | 0.9321 | 0.9213 | 0.9001 |
| MentalHealth | 0.9482 | 0.9318 | 0.9173 | 0.8966 |

**어떤 변수도 삭제하지 않았을 때,
평가 지표가 가장 좋았다**

# 3. 모델링

**2개의 상위모델(XGB, Catboost)을 선정 하고**
**하이퍼파라미터 튜닝 후에 두 모델을 소프트 보팅함**

- 소프트 보팅

## CATBOOST

bagging_temperature = 0, depth = 9,
l2_leaf_reg = 3,
learning_rate =0.1

| | train | test |
|---|---|---|
| 정확도 | 0.9336 -> 0.9532 | 0.9325 -> 0.9357 |
| 재현율 | 0.8933 -> 0.9240 | 0.8923 -> 0.9009 |

## XGBOOST

colsample_bytree = 0.8,
learning_rate = 0.1,
max_depth = 5,
max_leaf_nodes = 2,
min_child_weight = 4,
n_estimators = 200,
subsample = 0.9

| | train | test |
|---|---|---|
| 정확도 | 0.9366 -> 0.9562 | 0.9342-> 0.9435 |
| 재현율 | 0.8970 -> 0.9195 | 0.8946 -> 0.9065 |

## SOFT_VOTING(CAT & XGB)

| | train | test |
|---|---|---|
| 정확도 | 0.9580 | 0.9458 |
| 재현율 | 0.9207 | 0.9063 |

## 3. 모델링

- 모델 간 점수 비교

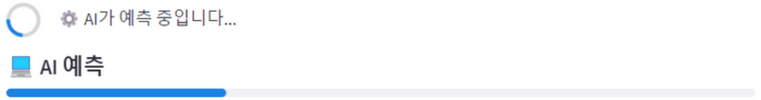
soft_voting의 평가지표가 유의미하게 높지만,
웹에 voting 모델을 로드하는 것에 기술적 한계가 있었음.
따라서 평가지표가 가장 좋은 xgboost로만 웹 구현

# 4. 웹 구현

- 웹 개요

Streamlit


사용자 정보 입력




모델 기반 예측


예측값 기반
시각화 자료 출력

## 4. 웹 구현

- 모델 웹 적용법 소개

- 웹 동작 때마다 머신러닝 모델이 데이터를 학습 해야만하는 문제

  -> 학습 시킨 모델을 저장하고, 웹 동작 시엔 그 모델을 불러오기만 하는 식으로 해결

1. joblib 라이브러리 활용
2. pkl 형식으로 학습된 모델 자체를 저장
3. 웹 동작시, 저장된 pkl 파일을 로드하는 방식으로 웹 동작

## 4. 웹 구현

- 모델 웹 적용법 소개


**Predict_Proba 활용 심장질환이 있을 확률**

**Predict 활용 심장질환 분류된 기준에 따라**

## 4. 웹 구현

- 모델 웹 적용법 소개


1. 사용자가 입력한 정보에, 일정 변수를 조작해서 predict_proba 값을 반환 받음

   (ex) 흡연을 했다고 응답한 응답자의 경우, 다른 조건은 일정한 상태에서 흡연을 하지 않은 걸로 바꾼 후의 예측 값

2. 사용자가 기존 입력한 정보와 기존 확률과 비교해서, 심장질환일 확률이 떨어졌다면 그 변수에 대해서 확률 출력

## 4. 웹 구현

- 웹 시연

## 5. 프로젝트 마무리

- 한계점

1. ML 하이퍼파라미터 튜닝을 시도하는데에 있어서**GPU 옵션**을 제공하지 않는 모델들이 존재하여 세밀한 조정에 대한 시도가 부족했다.
2. 보팅 모델을 웹에 구현하지 못한 것이 아쉬웠고, 분류기를 조금 더 여러 개 써서 앙상블을 시도 해봤으면 좋았을 것 같다. 특히 **스태킹**을 한번 해봤으면 어땠을까 싶었다.
3. **데이터베이스**를 활용했다면, 사용자가 입력한 정보를 데이터베이스에 입력하면서 조금 더 다양한 시도를 해볼 수 있었을 것 같다. ex) 이용 사용자 대시보드 생성, 심장질환 예측자 사후 서비스 등
4. 심장질환 관련 **의료기관을 지도 시각화**하여 심장질환이라고 분류된 사용자에게 병원 위치를 제공 하려고 했었다. 공공데이터 심질환 관련 의료기관 자료의 위치 좌표를 변경하는 과정에서 오차가 발생하여 시간이 너무 길어질 것 같아 제외했다.

## 5. 프로젝트 마무리

- 팀원별 느낀점

### 우상욱

- 머신러닝 모델이 웹에서 어떻게 효율적으로 동작하는지 이해했습니다.
- 성능이 좋은 AI 모델은 예측 결과 뿐만 아니라, 다양한 방식으로 새로운 인사이트를 줄 수 있을 것이라고 생각했습니다.

### 민병창

- 큰 데이터로 학습을 시켜보며 좋은 컴퓨터가 왜 필요한지 깨닫게 되는 시간이었습니다.
- 데이터 EDA에서 분석한 그대로 ML 학습이 이루어지는 모습을 보고 데이터가 인공지능 모델을 정하는 가장 중요한 요소라는 생각을 가지게 되었습니다.

### 김경목

- 변수들간의 상관관계를 나타내는 시각화 부분이 생각보다 어려웠다
- 머신러닝 모델을 이해하고 직접 경험해볼 수 있어서 뜻깊은 프로젝트가 되었다

## 5. 프로젝트 마무리

- 질의응답