[머신러닝 입문 트랙 시즌2]
심장 질환 예측 AI 해커톤

김윤

EDA

데이터 정보

	id	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	1	53	1	2	130	197	1	0	152	0	1.2	0	0	2	1
1	2	52	1	3	152	298	1	1	178	0	1.2	1	0	3	1
2	3	54	1	1	192	283	0	0	195	0	0.0	2	1	3	0
3	4	45	0	0	138	236	0	0	152	1	0.2	1	0	2	1
4	5	35	1	1	122	192	0	1	174	0	0.0	2	0	2	1

- id: 데이터 고유 id
- age: 나이
- sex: 성별 (여자 = 0, 남자 = 1)
- cp: 가슴 통증(chest pain) 종류
- 0 : asymptomatic 무증상
- 1 : atypical angina 일반적이지 않은 협심증
- 2 : non-anginal pain 협심증이 아닌 통증
- 3 : typical angina 일반적인 협심증
- trestbps: (resting blood pressure) 휴식 중 혈압(mmHg)
- chol: (serum cholestoral) 혈중 콜레스테롤 (mg/dl)
- fbs: (fasting blood sugar) 공복 중 혈당 (120 mg/dl 이하일 시 = 0, 초과일 시 = 1)
- restecg: (resting electrocardiographic) 휴식 중 심전도 결과
- 0: showing probable or definite left ventricular hypertrophy by Estes' criteria
- 1: 정상
- 2: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
- thalach: (maximum heart rate achieved) 최대 심박수
- exang: (exercise induced angina) 활동으로 인한 협심증 여부 (없음 = 0, 있음 = 1)
- oldpeak: (ST depression induced by exercise relative to rest) 휴식 대비 운동으로 인한 ST 하강
- slope: (the slope of the peak exercise ST segment) 활동 ST 분절 피크의 기울기
- 0: downsloping 하강
- 1: flat 평탄
- 2: upsloping 상승
- ca: number of major vessels colored by flouroscopy 형광 투시로 확인된 주요 혈관 수 (0~3 개)
- Null 값은 숫자 4로 인코딩됨
- thal: thalassemia 지중해빈혈 여부
- 0 = Null
- 1 = normal 정상
- 2 = fixed defect 고정 결함
- 3 = reversable defect 가역 결함
- target: 심장 질환 진단 여부
- 0: < 50% diameter narrowing
- 1: > 50% diameter narrowing

EDA

데이터 정보

결측치 존재하지 않음

하나를 제외한 모든 Feature의 Dtype이 int형임.

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 151 entries, 0 to 150
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   id        151 non-null    int64  
 1   age       151 non-null    int64  
 2   sex       151 non-null    int64  
 3   cp        151 non-null    int64  
 4   trestbps  151 non-null    int64  
 5   chol      151 non-null    int64  
 6   fbs       151 non-null    int64  
 7   restecg   151 non-null    int64  
 8   thalach   151 non-null    int64  
 9   exang     151 non-null    int64  
 10  oldpeak   151 non-null    float64
 11  slope     151 non-null    int64  
 12  ca        151 non-null    int64  
 13  thal      151 non-null    int64  
 14  target    151 non-null    int64  
dtypes: float64(1), int64(14)
memory usage: 17.8 KB
```

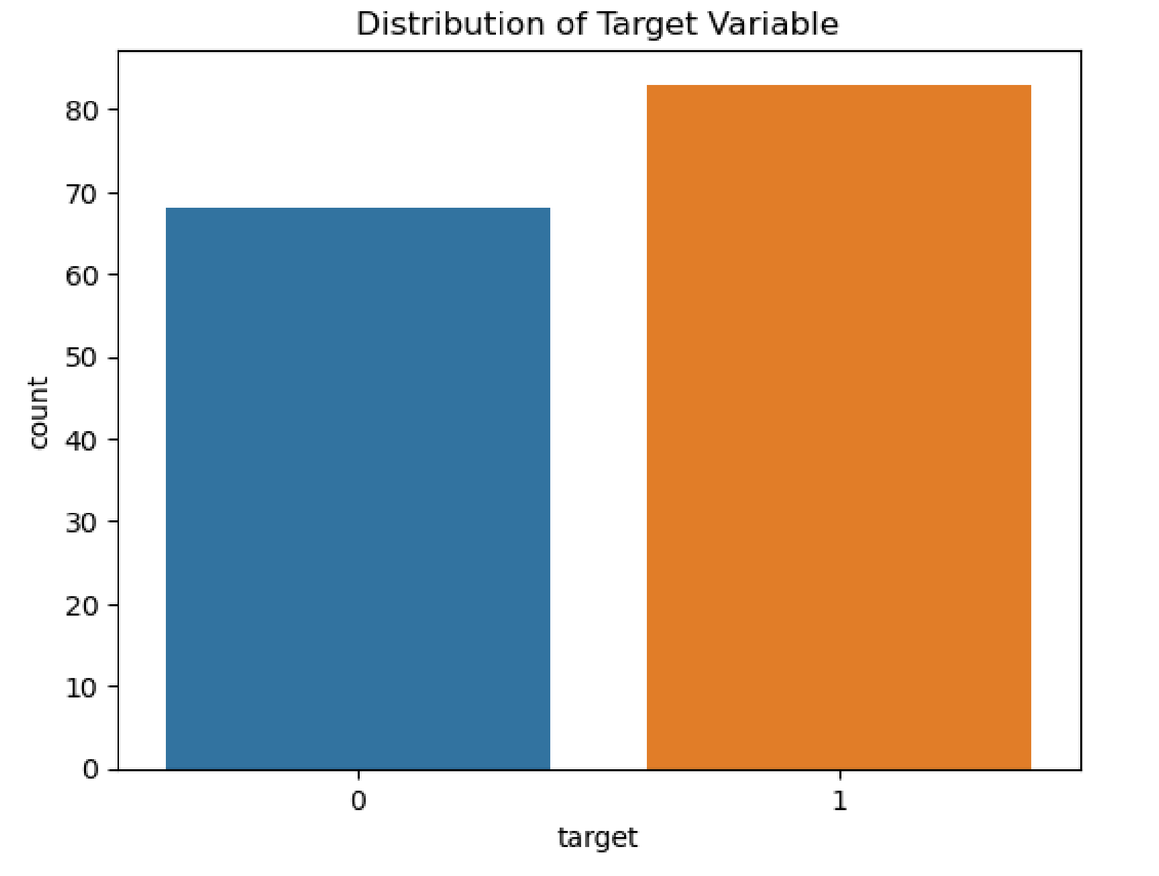
EDA

TARGET 분포 확인

심장 질환 여부(target)의 분포를 보여줌

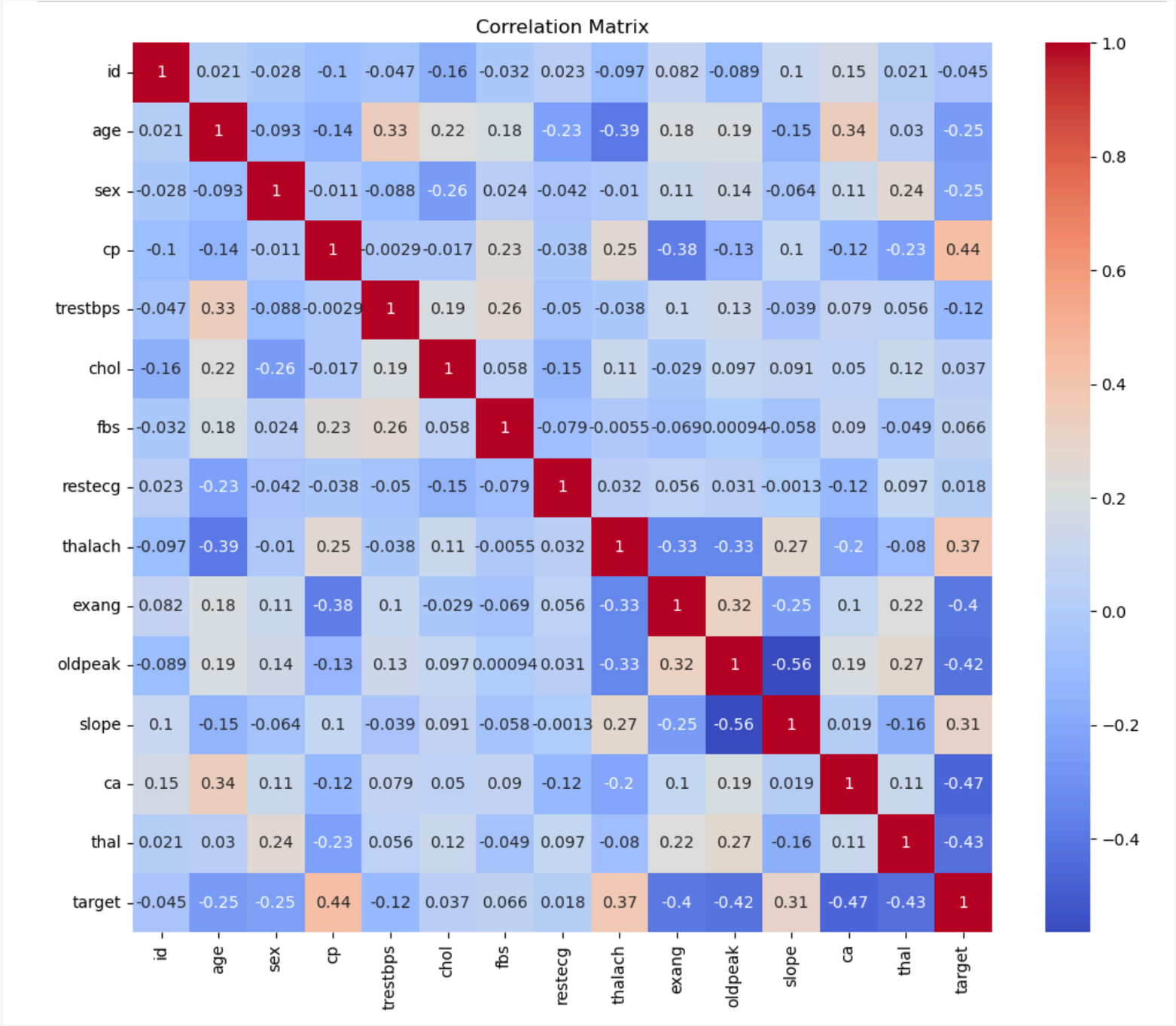
파란색 막대(target = 0) : 심장 질환이 없는 경우

주황색 막대(target = 1) : 심장 질환이 있는 경우



EDA

FEATURE 상관관계

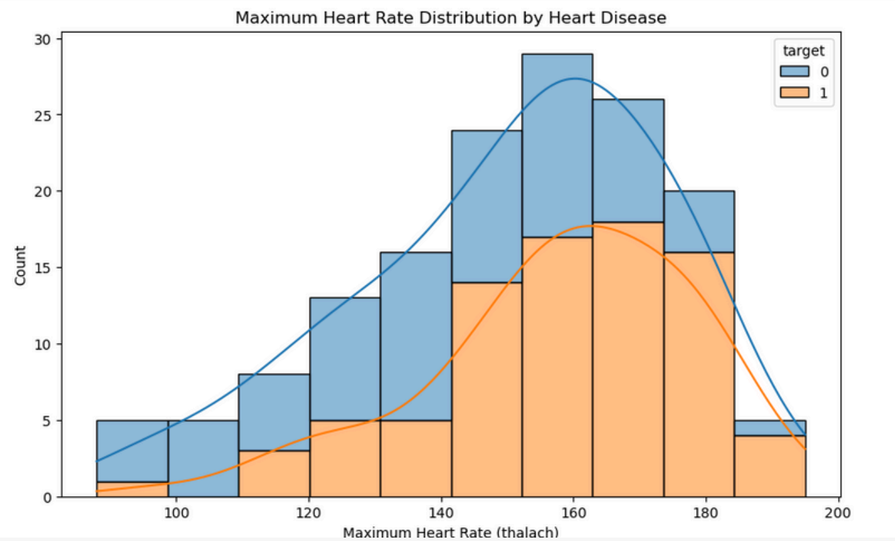


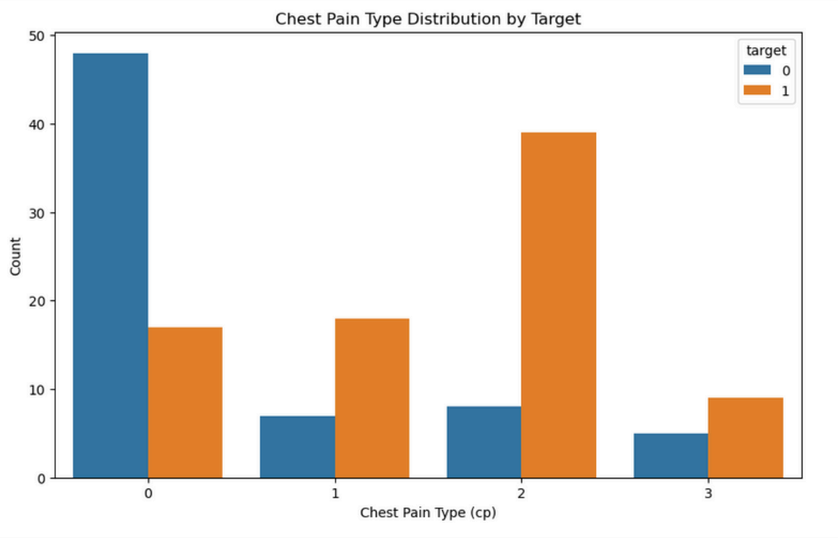
양의 상관관계 : cp-target / thalach-target

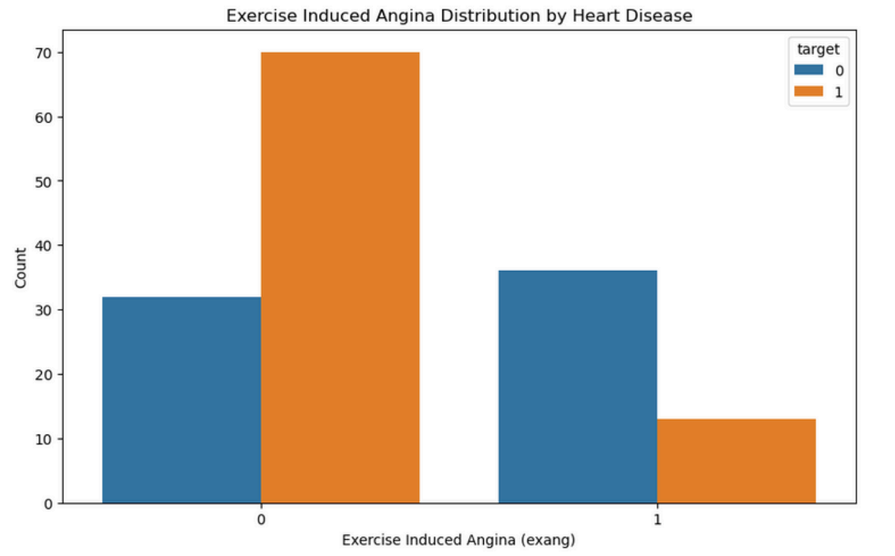
음의 상관관계 : exang-target / oldpeak-target / ca-target / thal-target

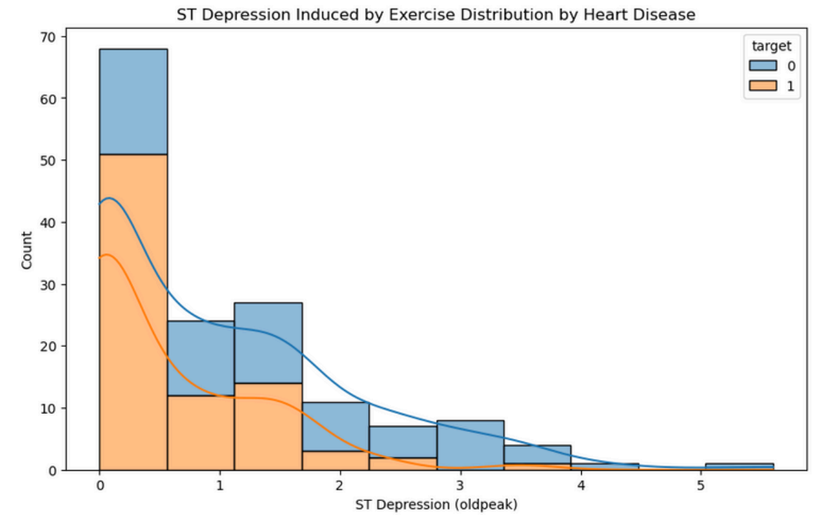
EDA

데이터 시각화


Maximum Heart Rate Distribution by Heart Disease
target
0
1
Count
Maximum Heart Rate (thalach)


Chest Pain Type Distribution by Target
target
0
1
Count
Chest Pain Type (cp)


Exercise Induced Angina Distribution by Heart Disease
target
0
1
Count
Exercise Induced Angina (exang)


ST Depression Induced by Exercise Distribution by Heart Disease
target
0
1
Count
ST Depression (oldpeak)

전처리

데이터 스케일링 및 분할

```
# 독립변수와 종속변수 분리
X = train_data.drop(columns=['id','target'])
y = train_data['target']

# 데이터 스케일링
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 학습 데이터와 테스트 데이터 분할
X_train, X_valid, y_train, y_valid = train_test_split(X_scaled, y, test_size = 0.2, random_state=42)
```

데이터 스케일링 이유 => 데이터셋의 각 특성은 서로 다른 범위와 단위를 가질 수 있다. 예를 들어 age는 34~77의 값을 가지고 있지만 chol은 수백 단위의 값을 가질 수 있다. 이러한 크기 차이 때문에, 크기가 큰 특성이 모델의 학습 과정에서 더 큰 영향을 미칠 수 있다. 이러한 영향을 줄이기 위해 각 특성의 값을 비슷한 범위로 조정한다.

모델링(학습 & 평가)

선택 모델 : Logistic Regression

```
# Logistic Regression 모델 생성 및 학습
model = LogisticRegression(max_iter = 5000)
model.fit(X_train,y_train)
```

선택 이유 :

1. 이진 분류 적합성 : 이 문제는 심장 질환이 있는지 없는 지를 예측하는 이진 분류 문제이다. Logistic Regression은 이러한 이진 분류 문제에 잘 맞는다.
2. 모델 단순성 : Logistic Regression은 비교적 간단한 모델로, 빠르게 학습할 수 있으며 과적합을 피할 수 있다.

```
# 모델 성능 평가
score = f1_score(y_valid, y_var_pred, average='micro')
print(score)
```

```
0.8387096774193549
```

=> f1 score 0.821 이상으로 목표를 달성했다.