# IMPORTING LIBRARIES

import numpy as np import pandas as pd import matplotlib. pyplot as PI t PI t. rcParams[ ' figure.figsize ' ] - [12, 7]

%matplotlib inline import seaborn as sns sns . set\_ style( "whitegrid " ) import statsmodels.api as sm from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LogisticRegression from sklearn.ensemble import RandomForestC1assifier from sklearn.metrics import accuracy\_ score, classification\_ report, confusion matrix #sns. set\_ context( "poster " ) #sns . set\_ context (None)

# READING DATA

In data pd . read\_excel ( " 1645792390\_cep1\_dataset . xlsx " )

## Perform preliminary data inspection

data. head()

Out [3] : age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target

o 63 1 3 145 233 1 o 150 o 2.3 o o 1 1

I 37 1 2 130 250 1 187 o 3.5 o o 2 1

1. 41  1 130 204 o 172 o 1.4 2 o 2 1
2. 56 1 1 120 236 1 178 o 0.8 2 o 2 1
3. 57 o o 120 354 1 163 1 0.6 2 o 2 1

In data. info()

<class 'pandas. core. frame.DataFrame'> Rangelndex: 303 entries, e to 302 Data columns (total 14 columns) :

 Column Non-Null Count Dtype



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | | --- | --- | --- | --- | | 1. age 2. sex 3. cp 4. trestbps 5. chol 6. fbs 7. restecg 8. thalach 9. exang 10. oldpeak 11. slope 12. ca 13. thal 14. target dtypes: float64(1), memory usage: | 303  303  303  303  303  303  303  303  303  303  303  303  303  303  33.3 | non-null non-null non-null non-null non-null non-null non-null non-null non-null non-null non-null non-null non-null non-null int64(13)  KB | int64 int64 int64 int64 int64 int64 int64 int64 int64 float64 int64 int64 int64 int64 | | data.isna() .sum() | |  |  | |

age 0

Out [5] :

sex 0

cp 0

trestbps 0

chol 0

fbs 0

restecg 0

thalach 0

exang 0

oldpeak 0

slope 0

ca 0

thal 0

target 0

dtype: int64

Get a preliminary statistical summary of the data

 data. describe()

Out [6] : age sex cp trestbps chol fbs restecg thalach exang oldpeak slope

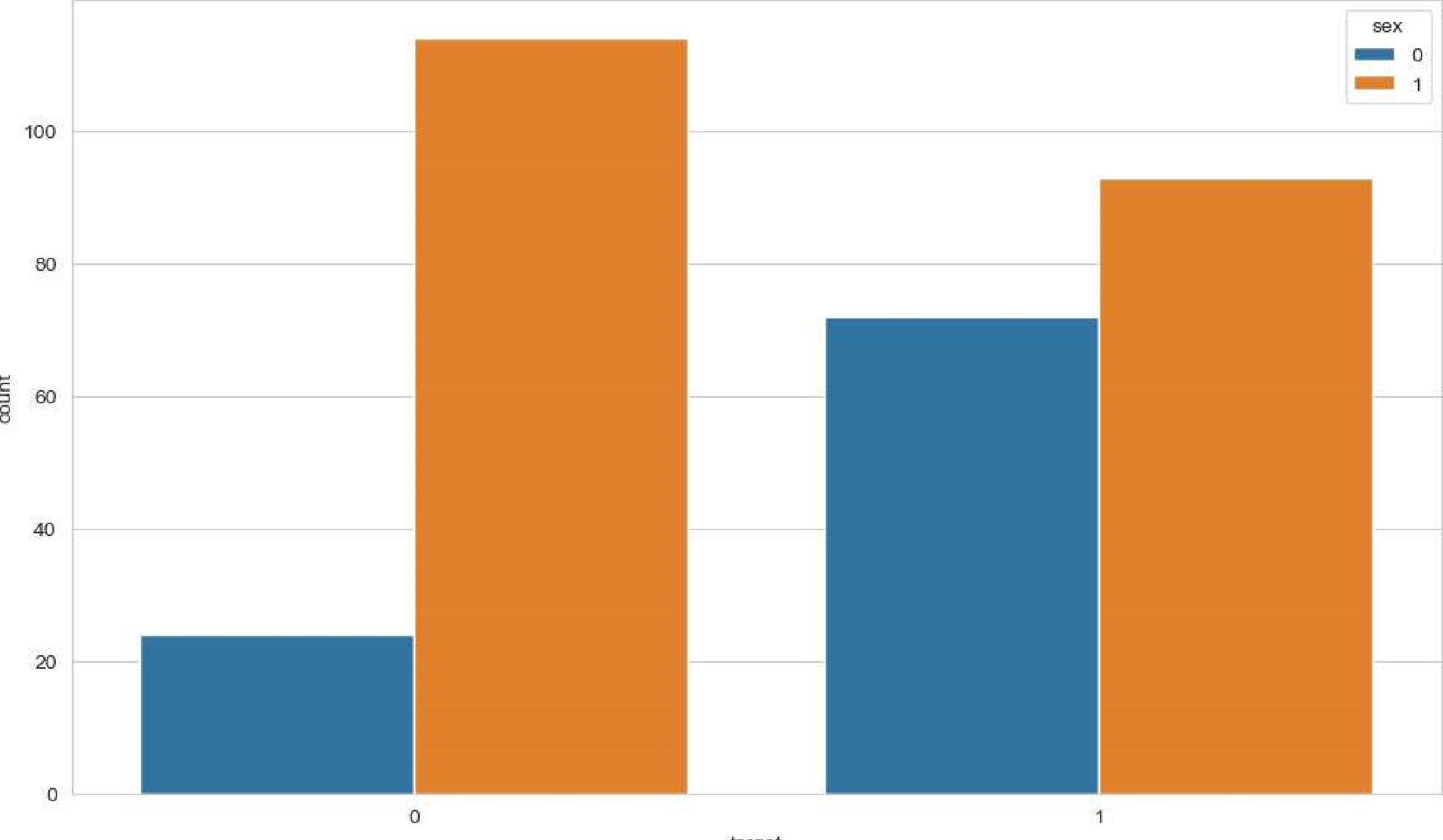
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| count | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 3 |
| mean | 54.366337 | 0.683168 | 0.966997 | 131.623762 | 246.264026 | 0.148515 | 0.528053 | 149.646865 | 0.326733 | 1.039604 | 1.399340 |  |
| std | 9.082101 | 0.46601 1 | 1.032052 | 17.538143 | 51.830751 | 0.356198 | 0.525860 | 22.905161 | 0.469794 | 1.161075 | 0.616226 |  |
| min | 29.000000 | 0.000000 | 0.000000 | 94.000000 | 126.000000 | 0.000000 | 0.000000 | 71.000000 | 0.000000 | 0.000000 | 0.000000 |  |
| 25% | 47.500000 | 0.000000 | 0.000000 | 120.000000 | 21 1.000000 | 0.000000 | 0.000000 | 133.500000 | 0.000000 | 0.000000 | 1.000000 |  |
| 50% | 55.000000 | 1.000000 | 1.000000 | 130.000000 | 240.000000 | 0.000000 | 1.000000 | 153.000000 | 0.000000 | 0.800000 | 1.000000 |  |
| 75% | 61.000000 | 1.000000 | 2.000000 | 140.000000 | 274.500000 | 0.000000 | 1.000000 | 166.000000 | 1.000000 | 1.600000 | 2.000000 |  |
| max | 77.000000 | 1.000000 | 3.000000 | 200.000000 | 564.000000 | 1.000000 | 2.000000 | 202.000000 | 1.000000 | 6.200000 | 2.000000 |  |



sns.countplot(data=data, 'target ' , hue=' sex' )

<AxesSubp10t: xlabel='target% ylabel=' count' >

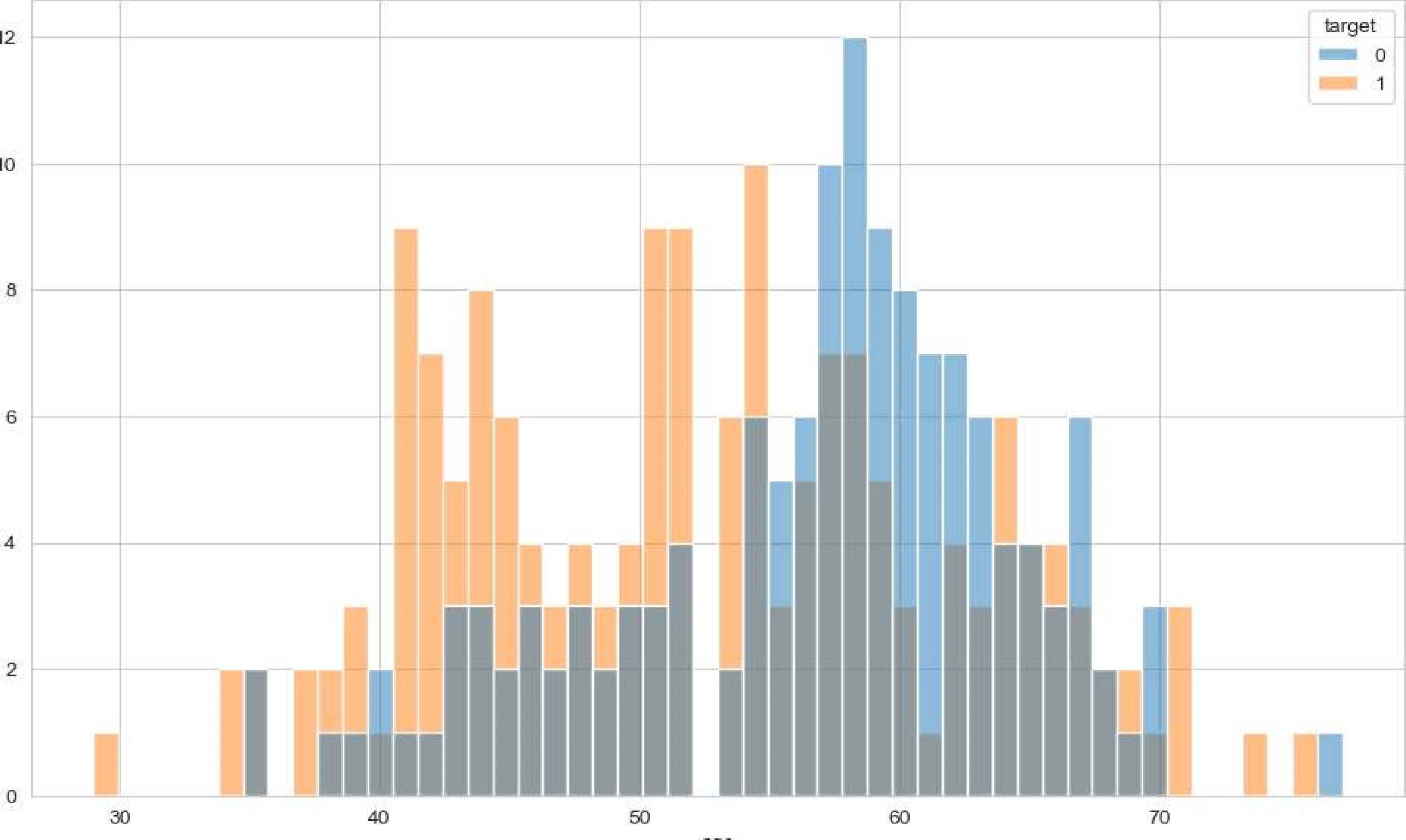
Out [7] :



ärget

In sns.histplot(data=data, age' bins=50, hue='target ' )

<AxesSubp10t: xlabel=' age' , ylabel= Count Out [8] :



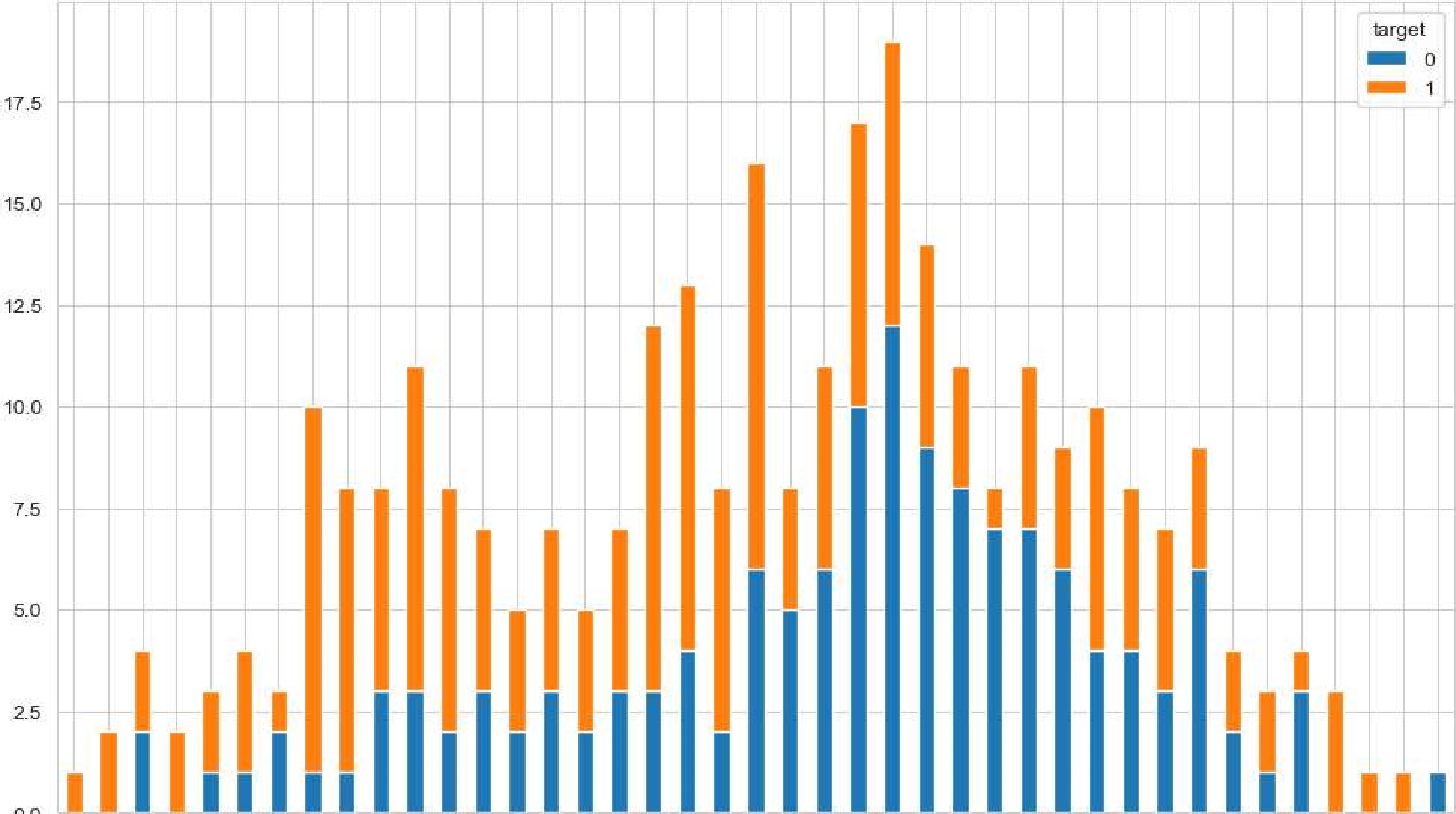
12

10

## age Study the occurrence of CVD across the Age category

age\_cvd\_counts=data . groupby( " age" ) [ ' target ' ] . value\_counts ( ) . unstack( ) age\_cvd\_counts . bar' , stacked=True)

<AxesSubp10t: xlabel=' age' > Out [9] :



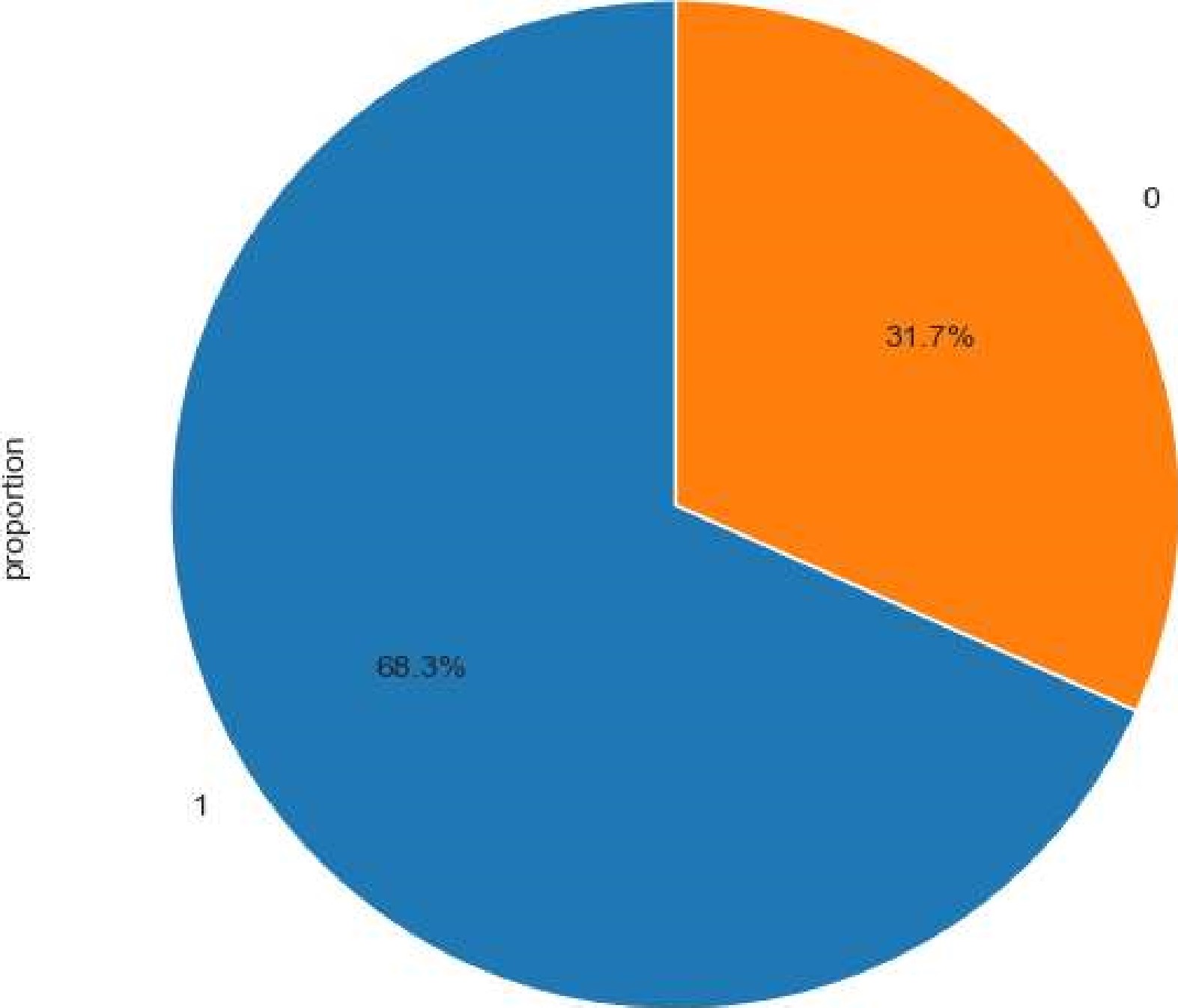
ege

Study the composition of all patients with respect to the Sex category

sex\_counts=data[ ' sex ' ] . value\_counts (normalize-True) sex\_counts . plot (kind= ' pie ' autopct= ' 0/01 . startang1e=90)

<AxesSubp10t: ylabel= ' proportion ' >

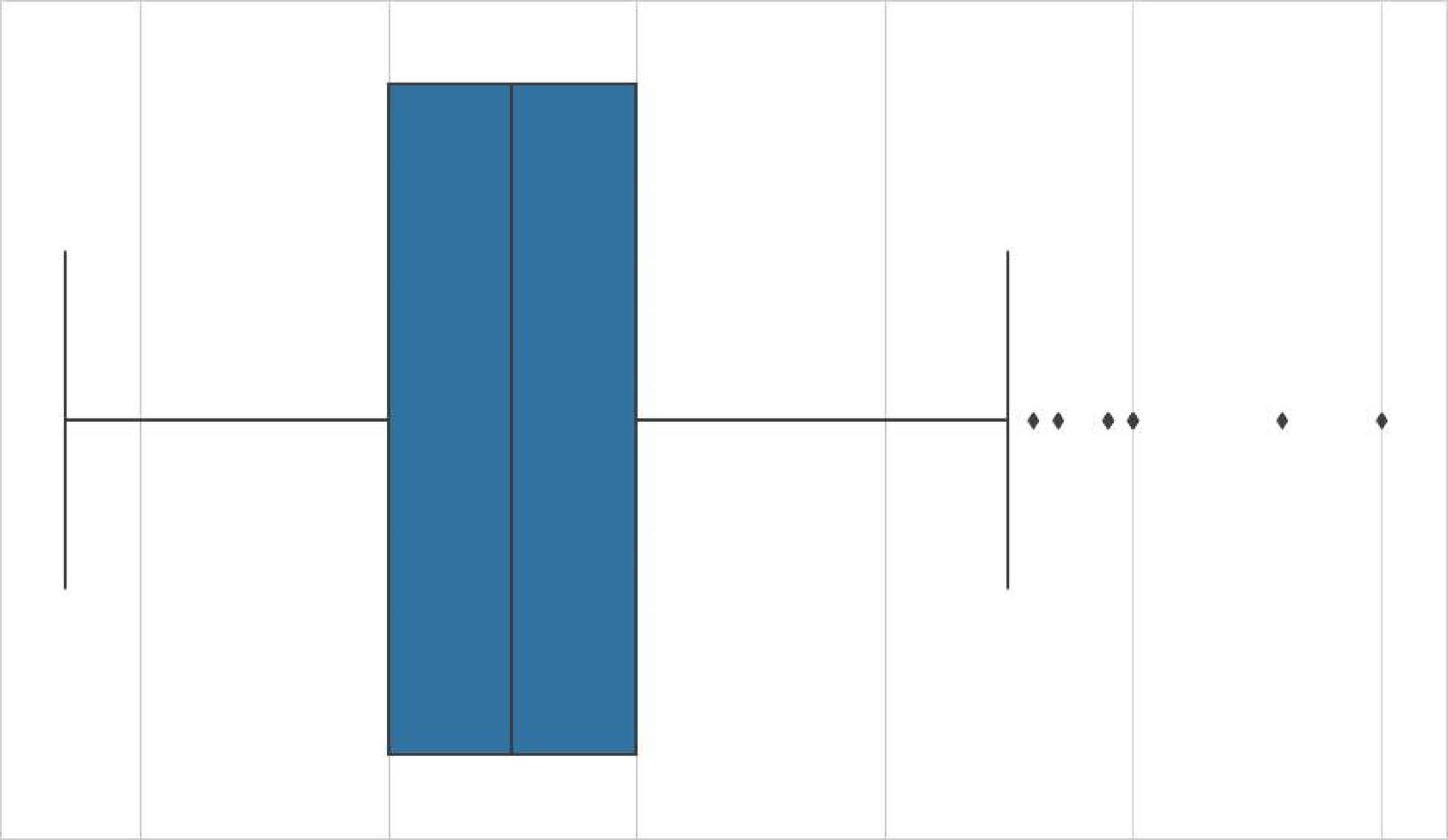
Out [10] :



## Study if one can detect heart attacks based on anomalies in the resting blood pressure (trestbps) of a patient

In [11] : sns.boxplot(data=data, x='trestbps ' , hue= ' target' )

Out[ll] : <AxesSubp10t: xlabel= 'trestbps ' >



100 120 440 460 180 200 testbps

anomalies  'trestbps ' ] < 90) | (datal 'trestbps' ] > 130)]

# Plot the resting blood pressure distribution

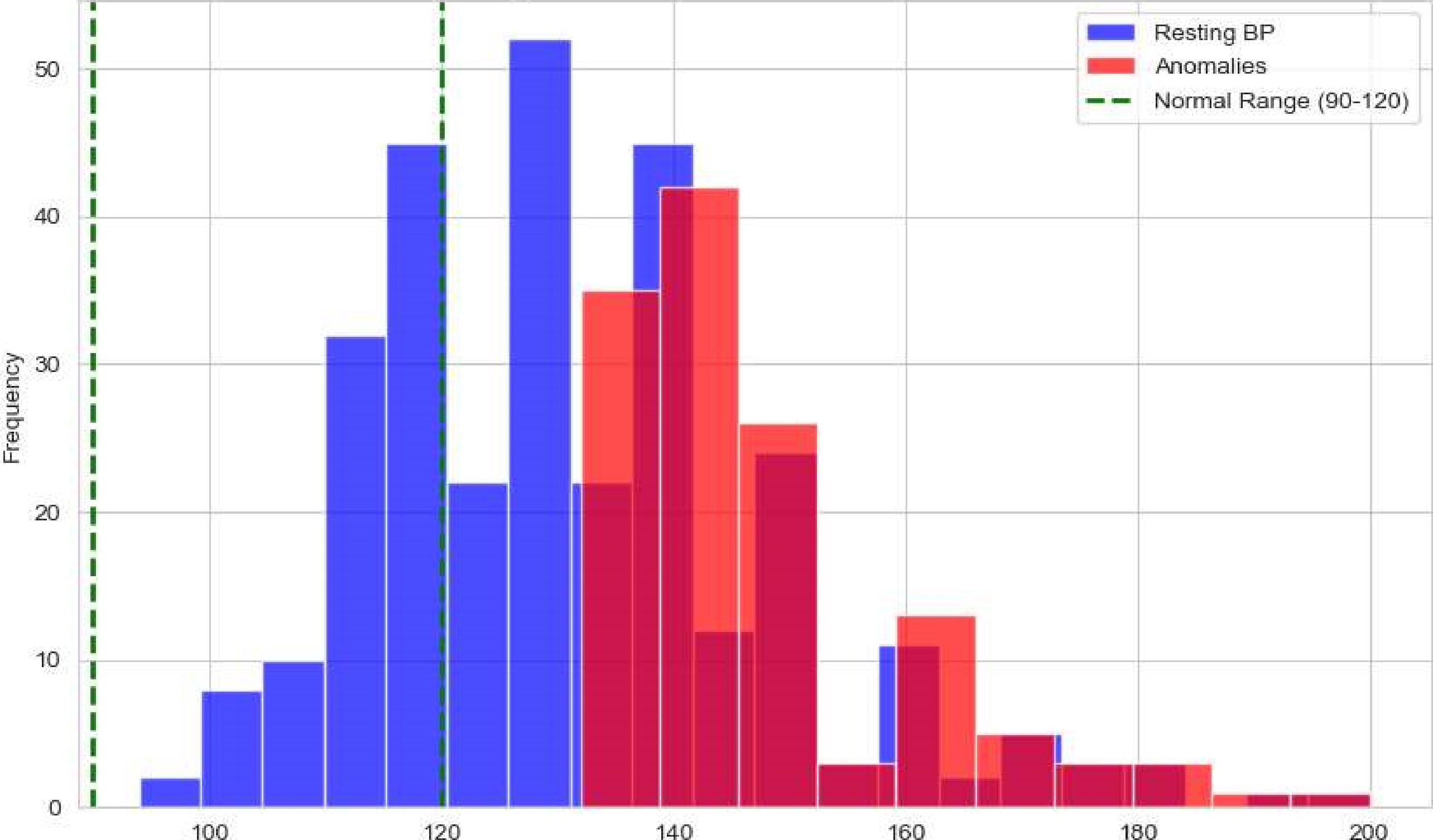


plt.hist(data[ 'trestbps ' ] , bins=2Ø, color='blue ' , alpha=e.7, label=' Resting BP I )

'trestbps ' ] , bins=10, color=' red ' , alpha=0.7, label- I Anomalies' ) plt.axv1ine(90, color='green ' linestyle= 'dashed ' , linewidth=2, label='Norma1 Range (90-120) ') plt.axv1ine(120, color=' green' , linestyle= ' dashed' , linewidth=2) plt.xlabel( 'Resting Blood Pressure (trestbps) plt.ylabel( ' Frequency' ) plt.title( ' Resting Blood Pressure Distribution with Anomalies ' )

PI t . legend() PI t . show()

Resting Blood Pressure Distribution with Anomalies



Resting Blood Pressure (trestbps)

List how the other factors determine the occurrence of CVD

In [13] : correlation\_matrix = data. corr() sns.heatmap(correlation\_matrix, annot=True, cmap= ' coolwarm' , linewidths=0.5)

Out [13] : <AxesSubp10t: >

Health Care\_Analysis

-0.09B 0069 0.28 0.21 0.12 o. 12 0.21 -0.17 0.28 0068



1.0

sex -o-oga -0.049 -0.057 -0.2 0045 -0.058 0.044 0.14 -0.031 0.12 0.21 -0.28

-0.049 0.048 -0.077 0.044 -0.15 0.12 -0.18 —0.16 0.43

trestbps 0.28 -0.057 0.048 0.12 0.18 -0.11 -0.047 -0.12 0.1 0062 -0.14

chol 0.21 -0.2 0.077 0.12 0.013 o. 15 -o.aogg 0.067 0.054 4.004 o.ogg o.0B5

0.12 0045 0.094 0.18 0,013 -o.os4 -0.0036 0 026 0.0057 0,14 -o. 032 0028

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| estecg | -0.12 | -o. 058 |  | -0.11 | -0.15 -o.oB4 | -0.07h | o. osg |  | -0.072 | -0.012 | 0.14 | 02 |
| flnalach |  | -0.044 | 0.3 | -0.047 | -o.oogg -0.0086 |  |  | 0.39 |  | -0.096 | 0.42 |  |

 exang 0.14 0.067 0.026 -0.071 0.29 0.12 -0 44

oddpeak 0.21 0096 0.15 0.19 0.054 0.0057 -0.059 0.29 -0.58 0.22 021 -0.43

slope 0.17 -0.031 0.12 0.12 -0.004 -0.06 ooga osg 0.35

ca 0,28 0.12 o. 16 0.1 0.071 0,14 -o. 072 0,12 0.22 -0.08 0.15

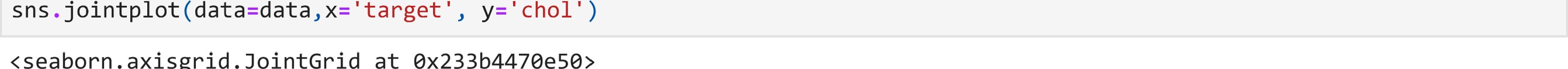
hal 0.21 -o. 16 0062 o.ogg -0.032 -0.012 -0.096 0.21 0.21 0.15

target 0.23 -0.28 0.39

age sex cp frestbps chol restecg thalach exang oddpeak dope thal target

## Describe the relationship between cholesterol levels and a target variable

In [14] :



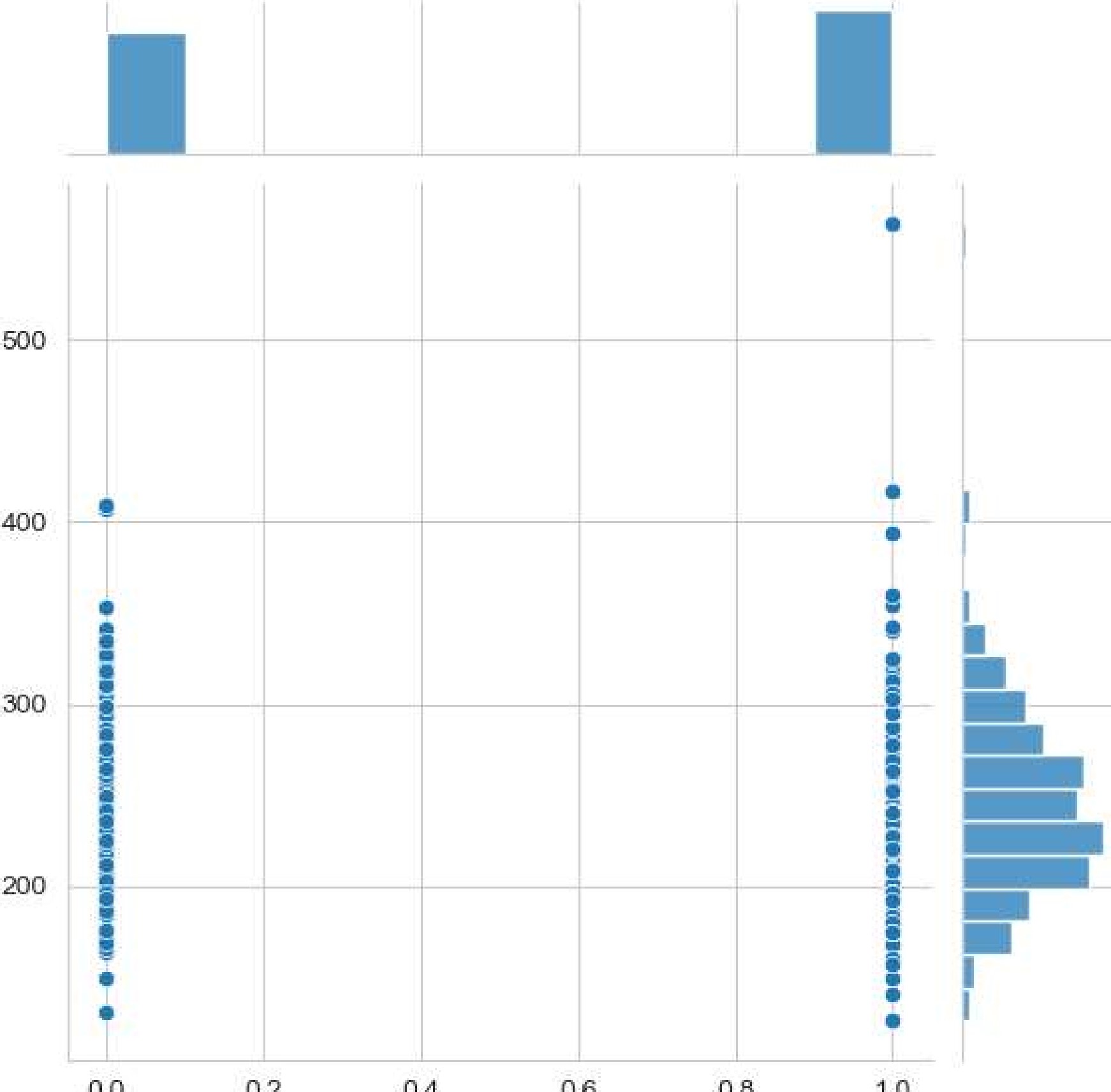
,

y:'

<seaborn.axisgrid.JointGrid

at

Out [14] :



0.0

0.2

0.4

0.6

0.8

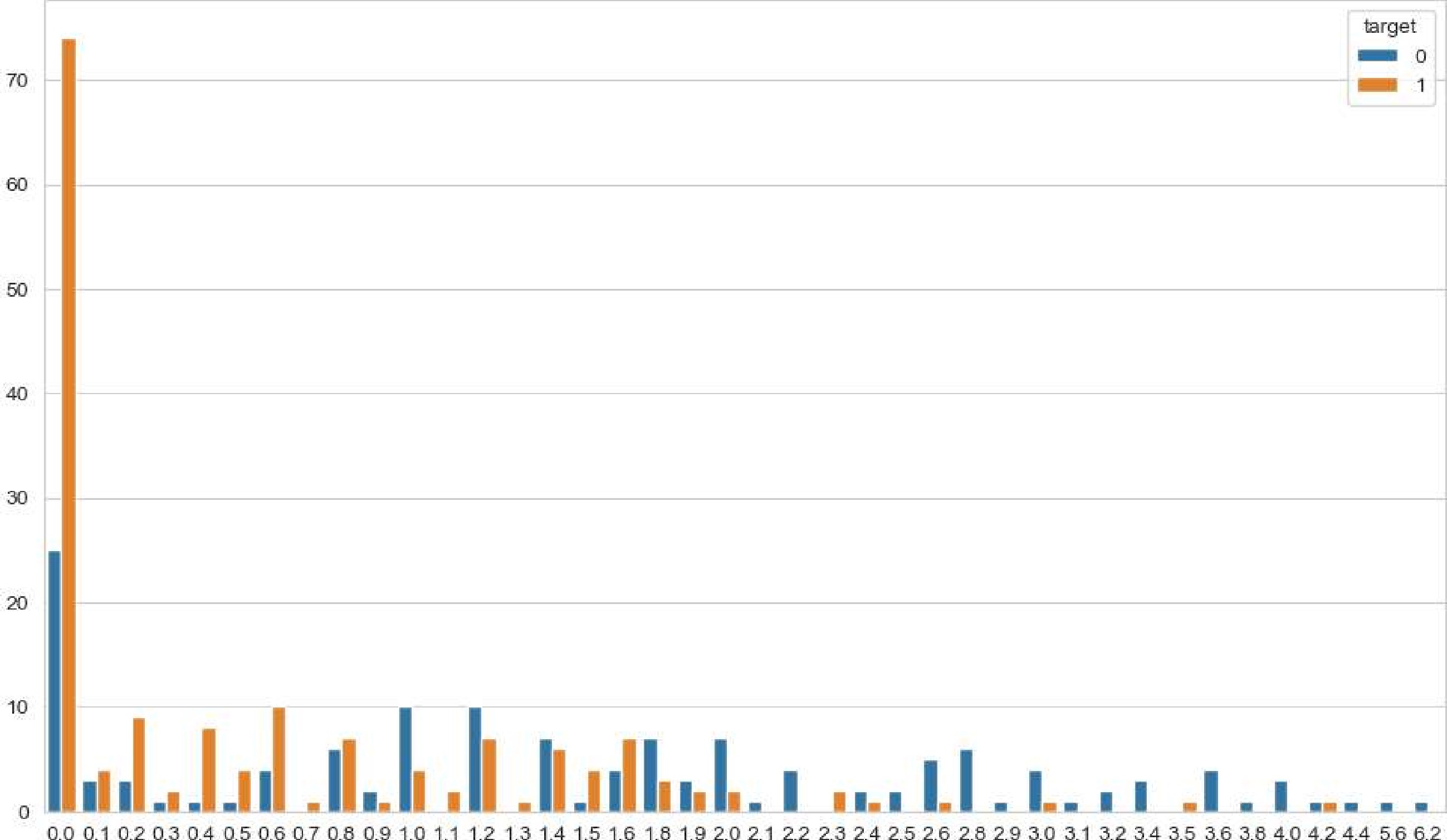
target

State what relationship exists between peak exercising and the occurrence of a heart attack

In [15] : sns.countplot(data=data, 'oldpeak' , hue= 'target ' )

<AxesSubp10t: xlabel='oldpeak' , ylabel=' count

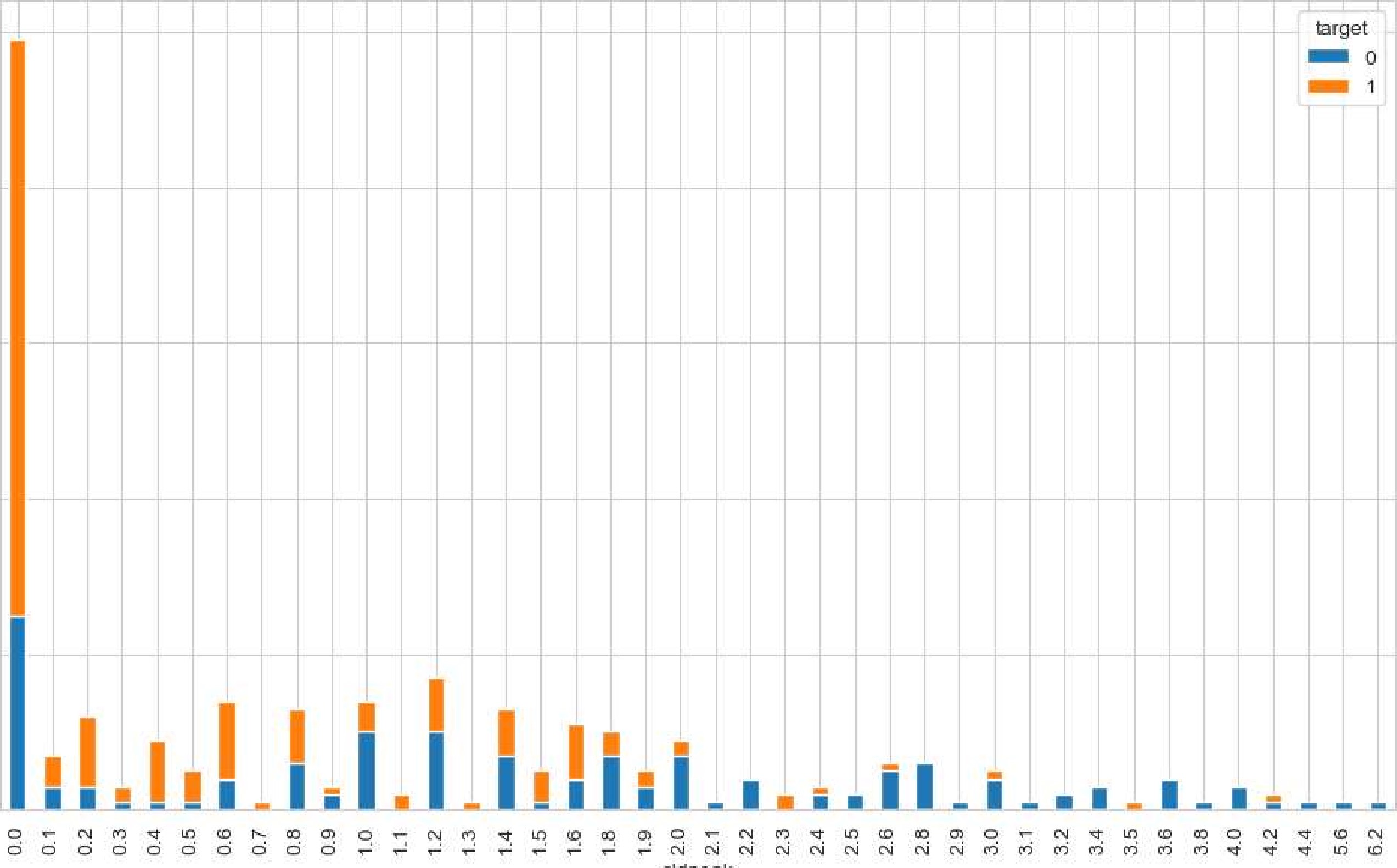
Out[15] :



Odpeak

oldpeak\_cvd\_counts=data . groupby ( "oldpeak" ) [ ' target ' ] . value\_ counts ( ) . unstack( ) oldpeak\_cvd\_counts  ' bar' , stacked-True)

<AxesSubp10t: xlabel= 'oldpeak' > Out[16] :



noo

cddpeak

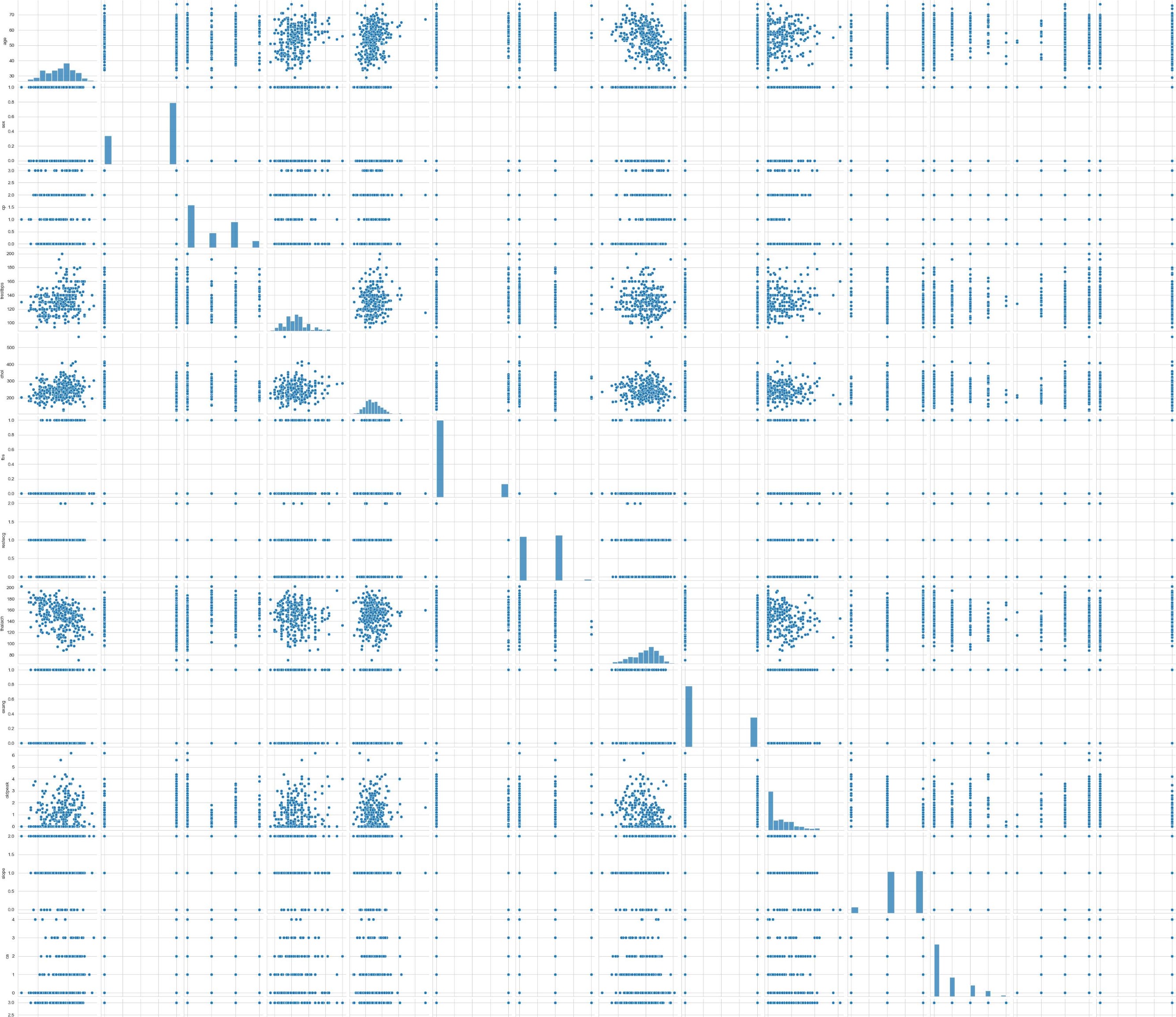
as we seen from above two graphs with low exercise there is high risk in CVD

PAIRPLOT FOR ALL VARIABLES

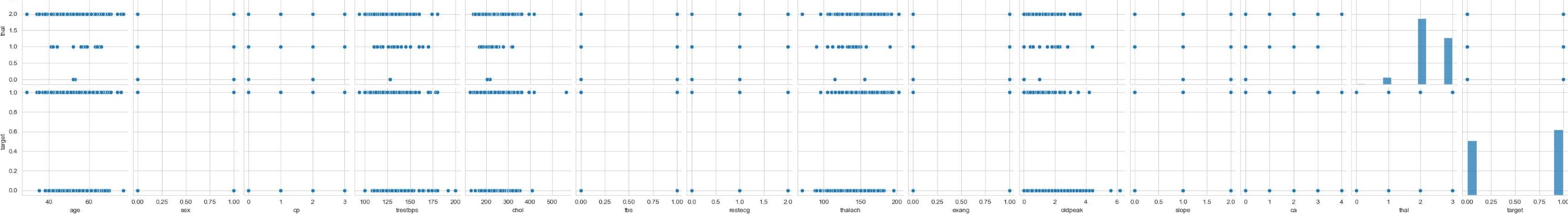
In [17] : sns.pairplot (data)

<seaborn.axisgrid.PairGrid at Ox233b453aa30>

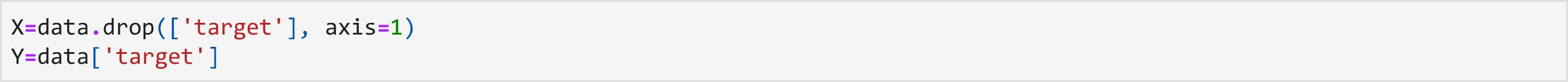
Out [17] :



localhost:8888/Iab/tree/HeaIth\_Care\_AnaIysis.ipynb



Build a baseline model to predict the risk of a heart attack using a logistic regression and random forest and explore the results while using correlation analysis and logistic regression (leveraging standard error and pvalues from statsmodels) for feature selection

In [18] : 

 sm. add \_constant (X)

Out [19] : const age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal

 1.0 63 145 233 150 2.3 o o 

 1.0 37 130 250 187 3.5 o o 

 1 .0 41 130 204 172  2 

 1.0 56 120 236 178 0.8 2 

 1.0 57 120 354 163 0.6 2 

1. 1.0 57 140 241 123 0.2 
2. 1110 264 132 1 .2 
3. 1.0 68 144 193 141 3.4 
4. 1.0 57 130 131 115 1 .2 
5. 1.0 57 130 236 174 0.0 
6. rows x 14 columns

In [20] : model=sm.OLS(Y,

localhost:8888/lab/tree/Health Care\_Analysis.ipynb

model . summary()

Out [21] : OLS Regression Results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dep. Variable: | | target R-squared (uncentered): | | | | | | 0.774 |
| Model: | | OLS Adj. R-squared (uncentered): | | | | | | 0.764 |
| Method: | | Least Squares | | |  | F-statistic: | | 76.48 |
| Date: | | Mon, 1 1 sep 2023 | | |  | Prob (F-statistic): | | 1.48e-85 |
| Time: | |  |  | |  | Log-LikeIihood: | | -1 12.42 |
| No. Observations: | |  | 303 | |  |  | AIC: | 250.8 |
| Df Residuals:  Df Model:  Covariance Type: | |  | 290  1 3 nonrobust | |  |  | BIC: | 299.1 |
|  | coef | std err | t | P > Itl | [0.025 | 0.9751 |  |  |
| age | 0.0031 | 0.002 |  | 0.180 | -0.001 | 0.008 |  |  |
| sex | -O. 1696 | 0.047 | -3.626 | o.ooo | -0.262 | -0.078 |  |  |
|  | 0.1 1 12 | 0.023 | 4.910 | o.ooo | 0.067 | 0.156 |  |  |
| trestbps | -0.0007 | 0.001 |  | 0.552 | -0.003 | 0.002 |  |  |
| chol | -O.OOOI | o.ooo | -0.316 | 0.752 | -0.001 | 0.001 |  |  |
| fbs | 0.0033 | 0.060 | 0.055 | 0.956 | -0.1 15 | o. 122 |  |  |
| restecg | 0.0712 | 0.040 | 1.794 | 0.074 | -0.007 | 0.149 |  |  |
| thalach | 0.0050 | 0.001 | 5.657 | o.ooo | 0.003 | 0.007 |  |  |
| exang | -0.1 194 | 0.051 | -2.331 | 0.020 | -0.220 | -0.019 |  |  |
| oldpeak | -0.0542 | 0.023 | -2.344 | 0.020 | -O. 100 | -0.009 |  |  |
| slope | 0.0888 | 0.043 | 2.077 | 0.039 | 0.005 | 0.173 |  |  |
|  | -0.1046 | 0.022 | -4.737 | o.ooo | -O. 148 | -0.061 |  |  |
| thal | -O. 1035 | 0.036 | -2.903 | 0.004 | -0.174 | -0.033 |  |  |
| Omnibus: | | 8.186 | Durbin-Watson: | | 1.046 | |  |  |

Prob(Omnibus): 0.017 Jarque-Bera (JB): 8.464

Skew: -0.407 Prob(JB): 0.0145 Kurtosis: 2.920 Cond. No. 961.

Notes:

1. R2 is computed without centering (uncentered) since the model does not contain a constant.
2. Standard Errors assume that the covariance matrix of the errors is correctly specified.

## As we saw from above summary that Prob (F-statistic) : 1.48e-85 which is less than standard p value 0.05. Hence the model is statistically significant

MODEL BUILDING AND EVALUATION

In [22] : X train, X test, y\_train, y\_test train\_test\_sp1it(X, Y, test\_size=e.2, random\_state=42)

## LOGISTIC REGRESSION

logistic\_model LogisticRegression() logistic\_model. fit (X\_train, y\_train) y\_pred\_logistic logistic\_model . predict (X\_test)

C : \Users \\_logistic . py : 458 : ConvergenceWarning: lbfgs failed to co nverge (status=l) :

STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max\_iter) or scale the data as shown in: https : //scikit-learn.org/stable/modules/preprocessing.html

Please also refer to the documentation for alternative solver options:

https : //scikit-learn.org/stable/modules/linear\_model . html#logistic - regression n iter i = \_check\_optimize\_result(

## RANDOM FOREST CLASSIFIER

|  |  |
| --- | --- |
| In [24] :  In [25] : | rf model - RandomForestC1assifier(n\_estimators=1ee, random\_state=42) rf\_mode1.fit(X\_train, y\_train) y\_ P red\_rf — rf\_model . predict (X\_test )  print("Logistic Regression:  print( "Accuracy: " accuracy\_score(y\_test, y\_pred\_logistic)) print(classification\_report(y\_test, y\_pred\_logistic)) print(confusion\_matrix(y\_test, y\_pred\_logistic)) |

Logistic Regression:

Accuracy: e. 8852459ô16393442 precision recall fl-score support

e. 89 e. 86 e.88 29

1 e.88 6.91 e. 89 32

accuracy 6.89 61 macro avg e. 89 e. 88 e. 88 61 weighted avg 6.89 6.89 e. 89 61

[ [25 4]

[ 3 29] ]

In [26] : print("\nRandom Forest:

print( "Accuracy: " accuracy\_score(y\_test, y\_pred\_rf) ) print(classification\_report(y\_test, y\_pred\_rf)) print(confusion\_matrix(y\_test, y\_pred\_

Random Forest:

Accuracy: e.836Ê6557377ô4918 precision recall fl-score support

e. 83 6.83 e.83 29 1 e. 84 e. 84 ô. 84 32

accuracy e. 84 61 macro avg e. 84 e. 84 e. 84 61 weighted avg e. 84 e. 84 e. 84 61

[ [24 5]

[ 5 27] ]