Course-End Project: Healthcare

Problem statement:

- Cardiovascular diseases are the leading cause of death globally. It is therefore necessary to identify the causes and develop a system to predict heart attacks in an effective manner. The data below has the information about the factors that might have an impact on cardiovascular health.

Dataset description:

- Variable - Description
- Age - Age in years
- Sex - 1 = male; 0 = female
- cp - Chest pain type
- trestbps - Resting blood pressure (in mm Hg on admission to the hospital)
- chol - Serum cholesterol in mg/dl
- fbs - Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- restecg - Resting electrocardiographic results
- thalach - Maximum heart rate achieved
- exang - Exercise induced angina (1 = yes; 0 = no)
- oldpeak - ST depression induced by exercise relative to rest
- slope - Slope of the peak exercise ST segment
- ca - Number of major vessels (0-3) colored by fluoroscopy
- thal - 3 = normal; 6 = fixed defect; 7 = reversible defect
- Target - 1 or 0

Task to be performed:

1. Preliminary analysis:

- a. Perform preliminary data inspection and report the findings on the structure of the data, missing values, duplicates, etc.
- b. Based on these findings, remove duplicates (if any) and treat missing values using an appropriate strategy

2. Prepare a report about the data explaining the distribution of the disease and the related factors using the steps listed below:

- a. Get a preliminary statistical summary of the data and explore the measures of central tendencies and spread of the data
- b. Identify the data variables which are categorical and describe and explore these variables using the appropriate tools, such as count plot
- c. Study the occurrence of CVD across the Age category
- d. Study the composition of all patients with respect to the Sex category

- e. Study if one can detect heart attacks based on anomalies in the resting blood pressure (trestbps) of a patient
- f. Describe the relationship between cholesterol levels and a target variable
- g. State what relationship exists between peak exercising and the occurrence of a heart attack
- h. Check if thalassemia is a major cause of CVD
- i. List how the other factors determine the occurrence of CVD
- j. Use a pair plot to understand the relationship between all the given variables

3. 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 p-values from statsmodels) for feature selection

Import Libraries

In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

In [2]:

```
url = 'https://docs.google.com/spreadsheets/d/1dNK1eyw3ljTT_oNTl8LDDLULzIbT6X80/
df = pd.read_excel(url)
```

1. Preliminary analysis

In [3]:

```
# Check the structure of the data
df.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	

In [4]:

```
df.shape
```

Out[4]:

```
(303, 14)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [6]:
```
df.isna().sum()
```

Out[6]:
```
age         0
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
```

In [7]:
```
# Check for duplicates
df.duplicated().sum()
```

Out[7]:
```
1
```

In [8]:
```
df = df.drop_duplicates()
```

In [9]:
```
df.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
df.shape
```

Out[10]:
```
(302, 14)
```

2. Prepare a report about the data

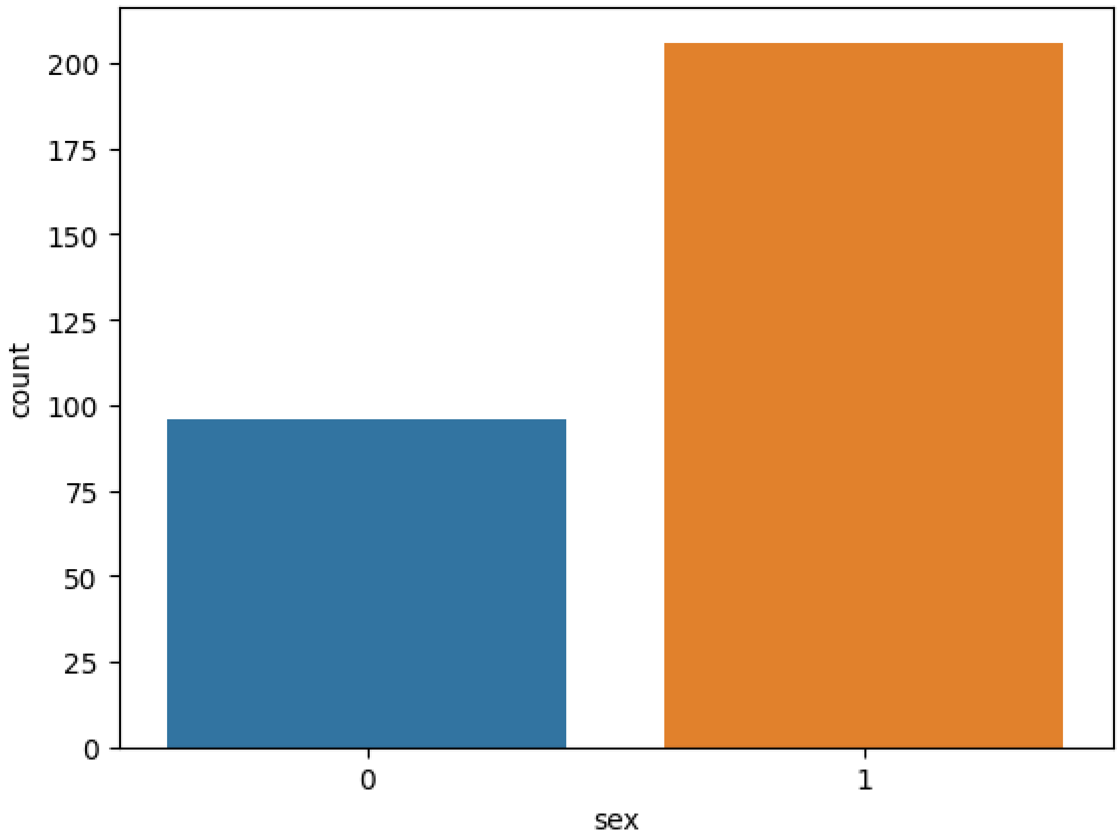
In [11]:

```python
# a. Preliminary statistical summary
print(df.describe())
```

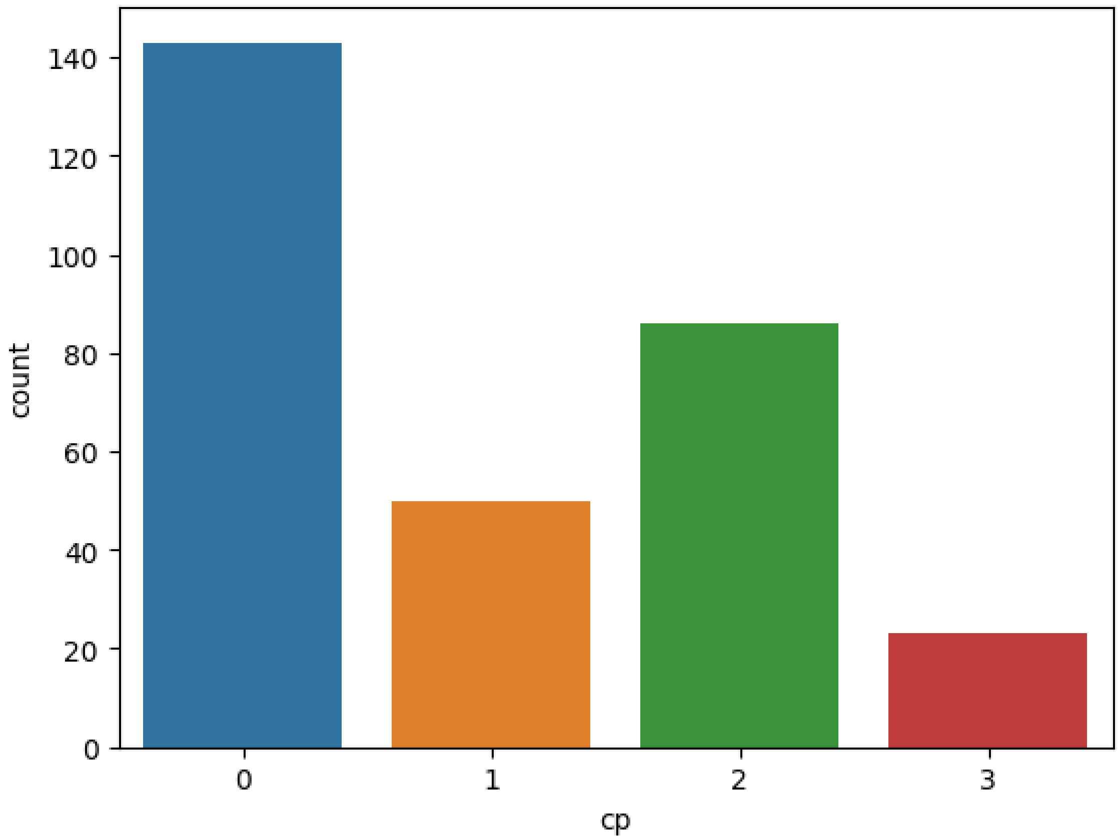
```
              age         sex          cp    trestbps        chol         fbs  \
count  302.00000  302.000000  302.000000  302.000000  302.000000  302.000000   
mean    54.42053    0.682119    0.963576  131.602649  246.500000    0.149007   
std      9.04797    0.466426    1.032044   17.563394   51.753489    0.356686   
min     29.00000    0.000000    0.000000   94.000000  126.000000    0.000000   
25%     48.00000    0.000000    0.000000  120.000000  211.000000    0.000000   
50%     55.50000    1.000000    1.000000  130.000000  240.500000    0.000000   
75%     61.00000    1.000000    2.000000  140.000000  274.750000    0.000000   
max     77.00000    1.000000    3.000000  200.000000  564.000000    1.000000   

          restecg     thalach       exang     oldpeak       slope          ca  
\
count  302.000000  302.000000  302.000000  302.000000  302.000000  302.000000   
mean     0.526490  149.569536    0.327815    1.043046    1.397351    0.718543   
std      0.526027   22.903527    0.470196    1.161452    0.616274    1.006748   
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000   
25%      0.000000  133.250000    0.000000    0.000000    1.000000    0.000000   
50%      1.000000  152.500000    0.000000    0.800000    1.000000    0.000000   
75%      1.000000  166.000000    1.000000    1.600000    2.000000    1.000000   
max      2.000000  202.000000    1.000000    6.200000    2.000000    4.000000   

             thal      target  
count  302.000000  302.000000  
mean     2.314570    0.543046  
std      0.613026    0.498970  
min      0.000000    0.000000  
25%      2.000000    0.000000  
50%      2.000000    1.000000  
75%      3.000000    1.000000  
max      3.000000    1.000000  
```

In [15]:

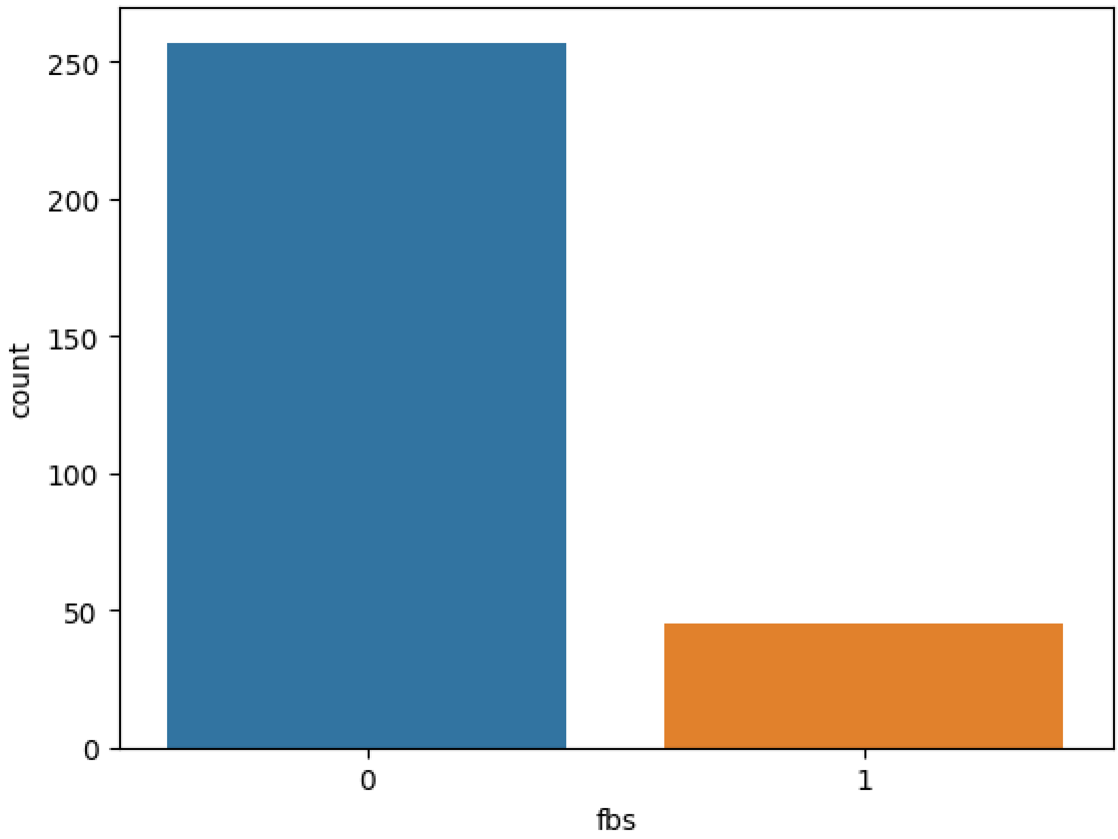
```python
# b. Categorical variables exploration
categorical_vars = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal
for var in categorical_vars:
    sns.countplot(x=var, data=df)
    plt.show()
```

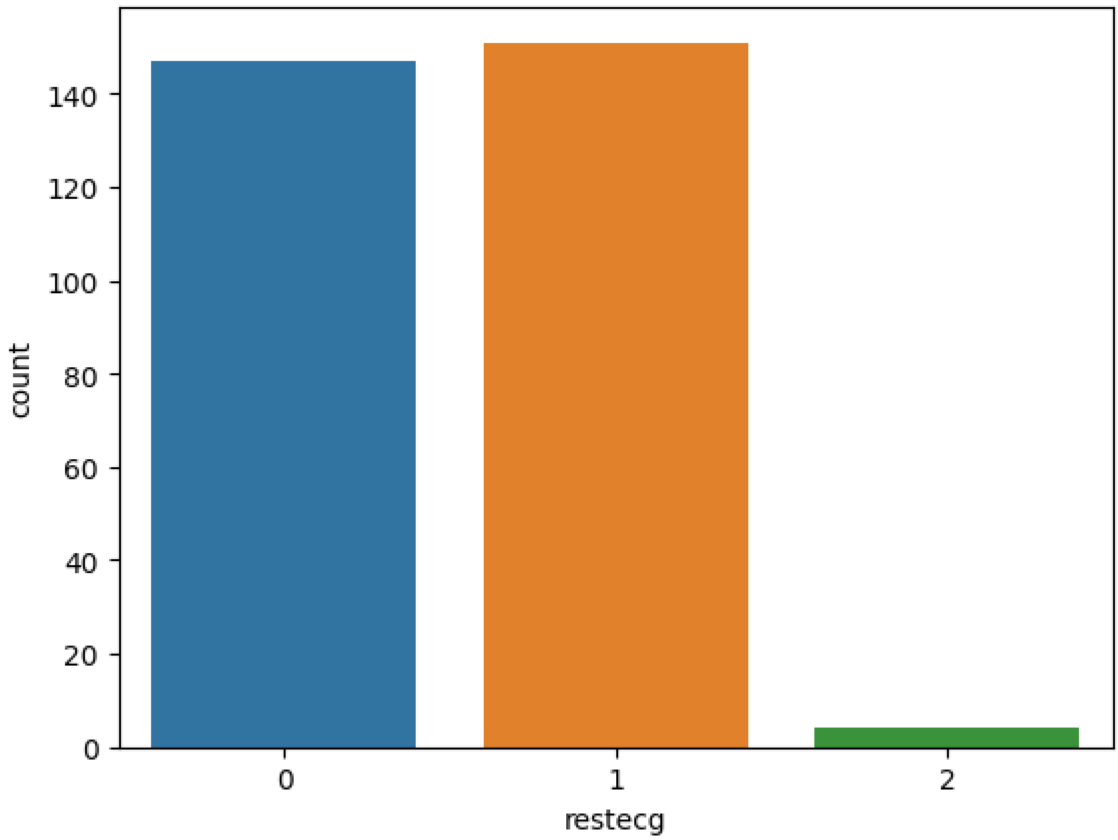


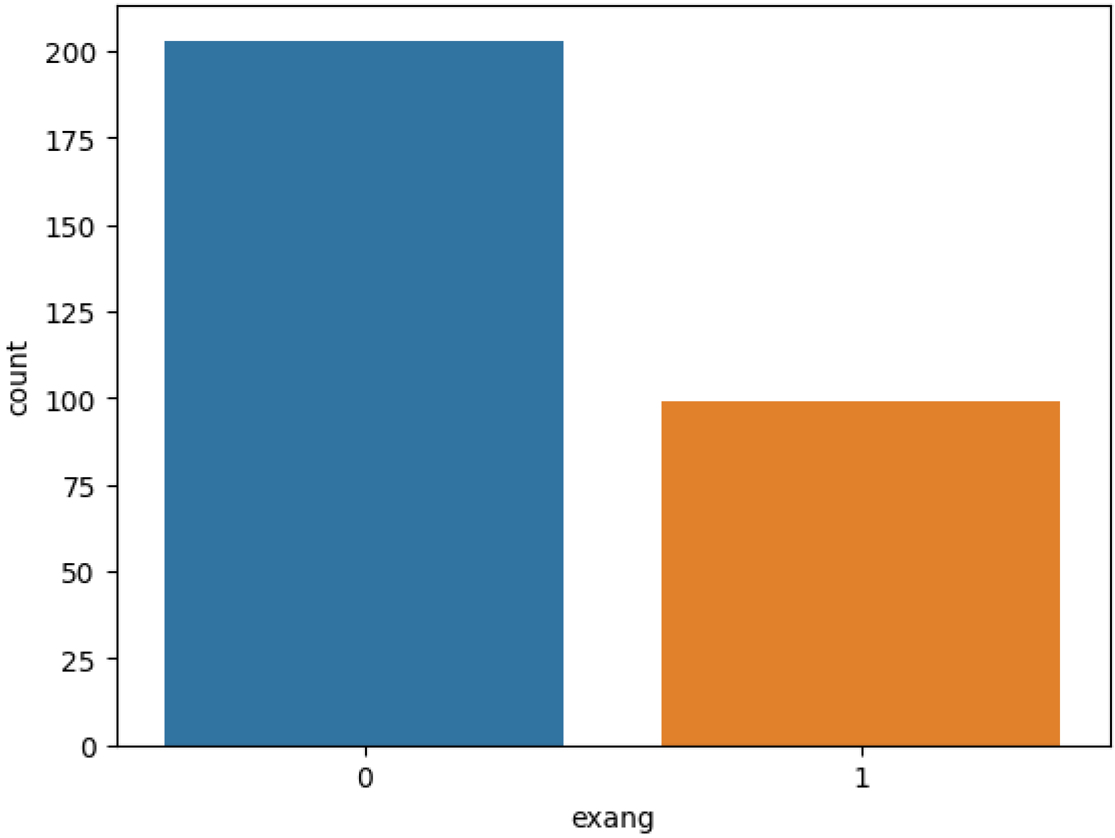
count
sex
0
1
0
25
50
75
100
125
150
175
200

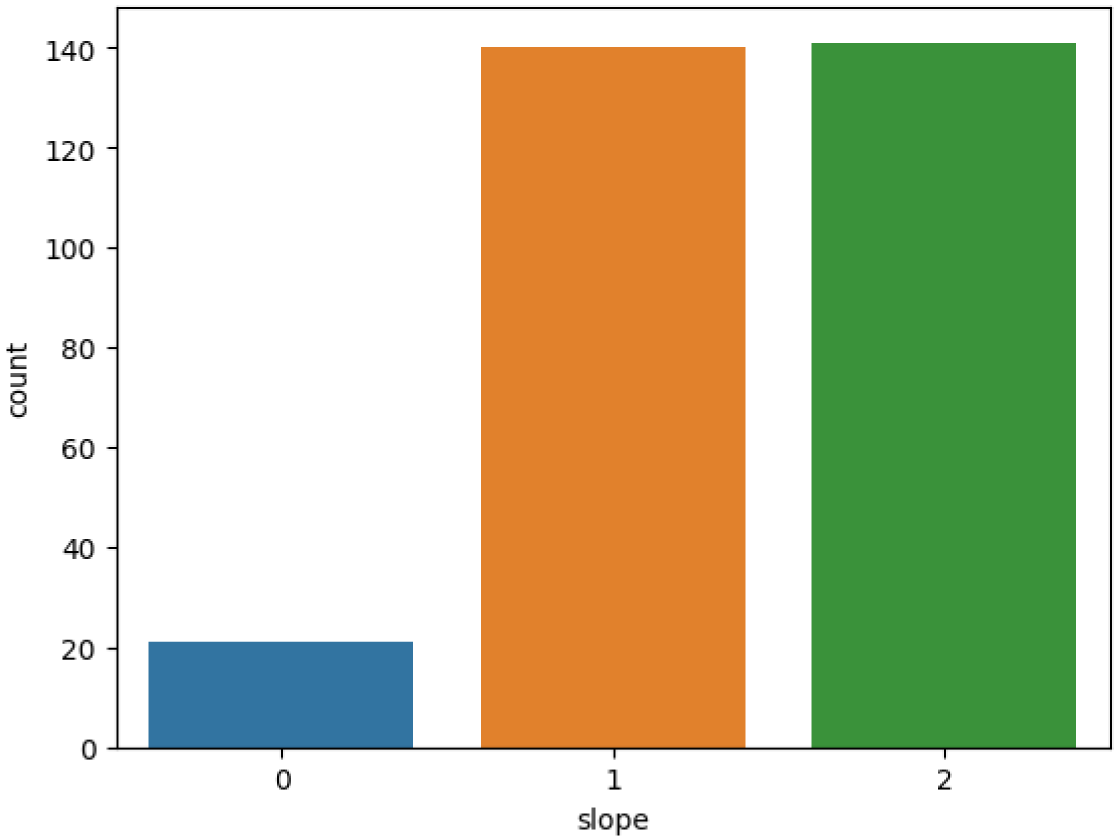


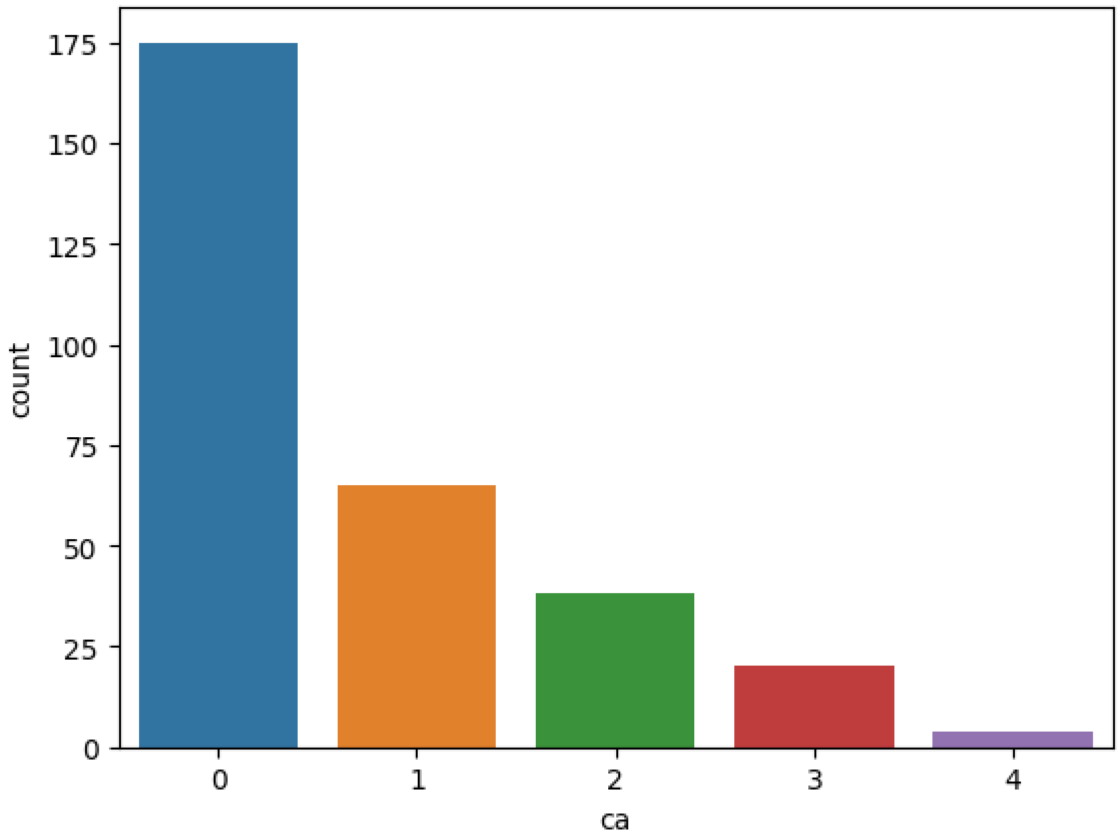
count
cp
0
1
2
3
0
20
40
60
80
100
120
140


count
250
200
150
100
50
0
0
1
fbs

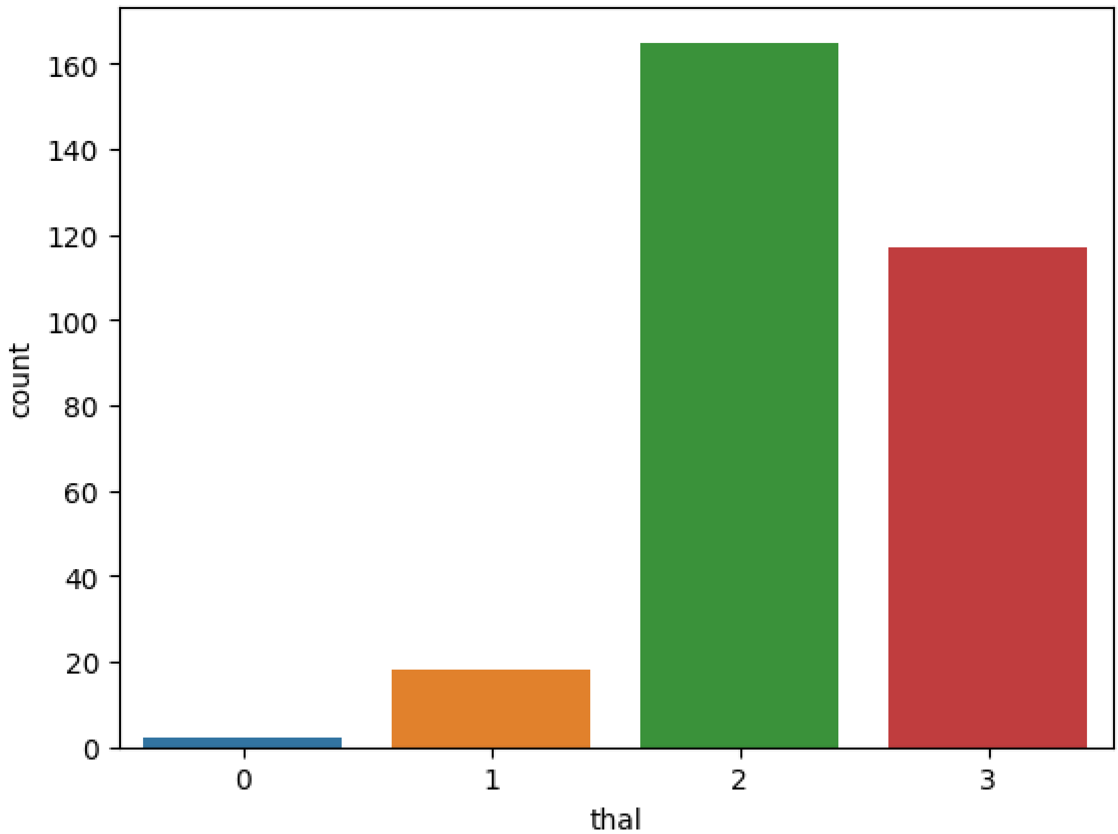

count
140
120
100
80
60
40
20
0
0
1
2
restecg


200
175
150
125
100
75
50
25
0
count
0
1
exang

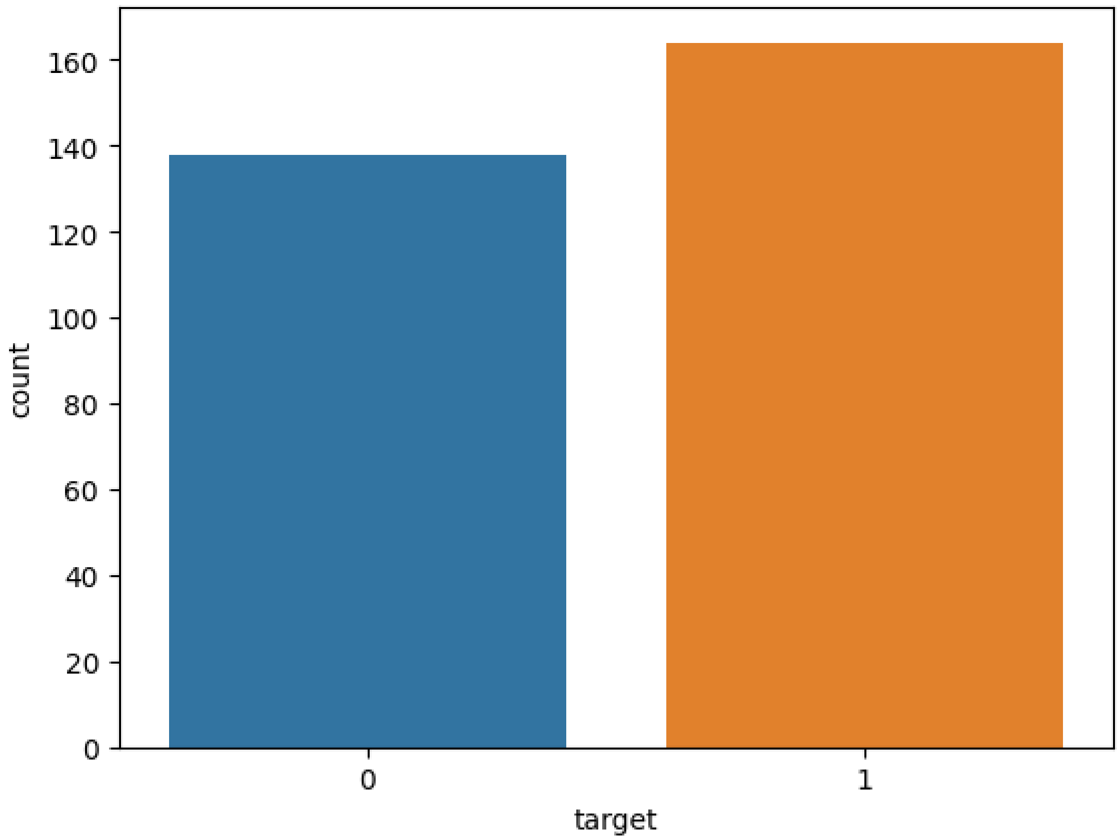

140
120
100
80
60
40
20
0
count
0
1
2
slope



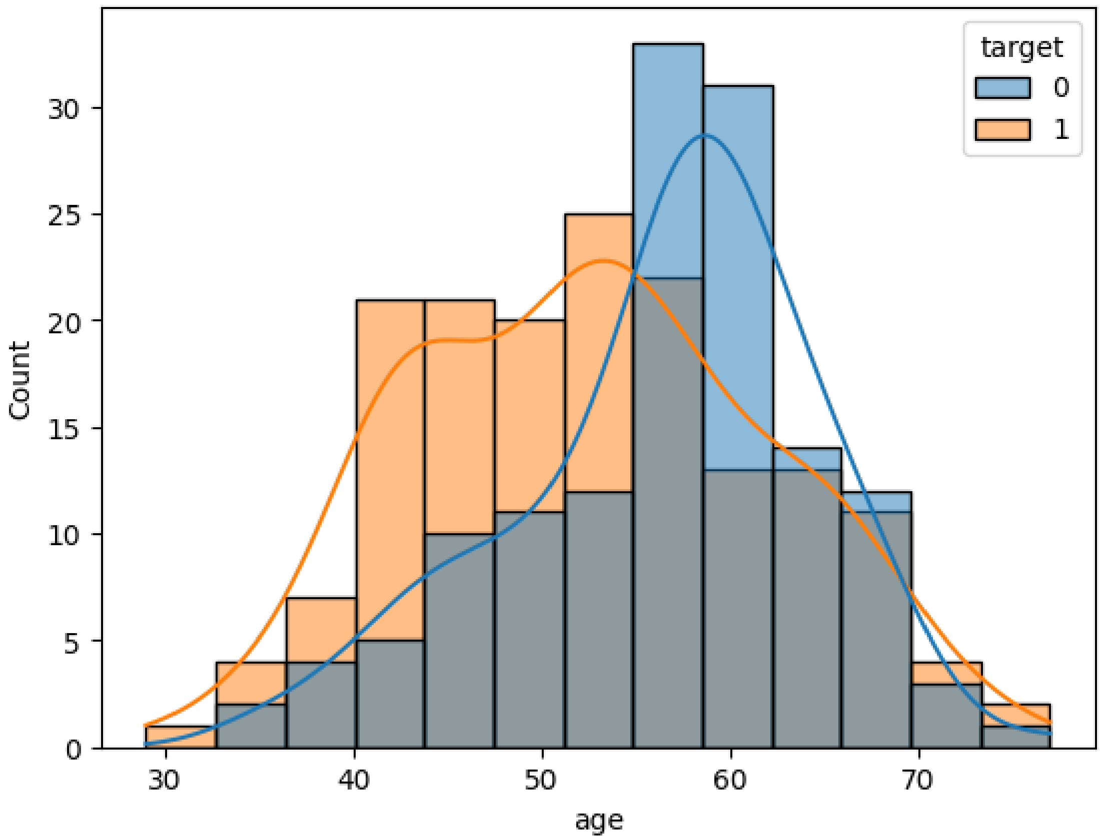
count
175
150
125
100
75
50
25
0
0
1
2
3
4
ca



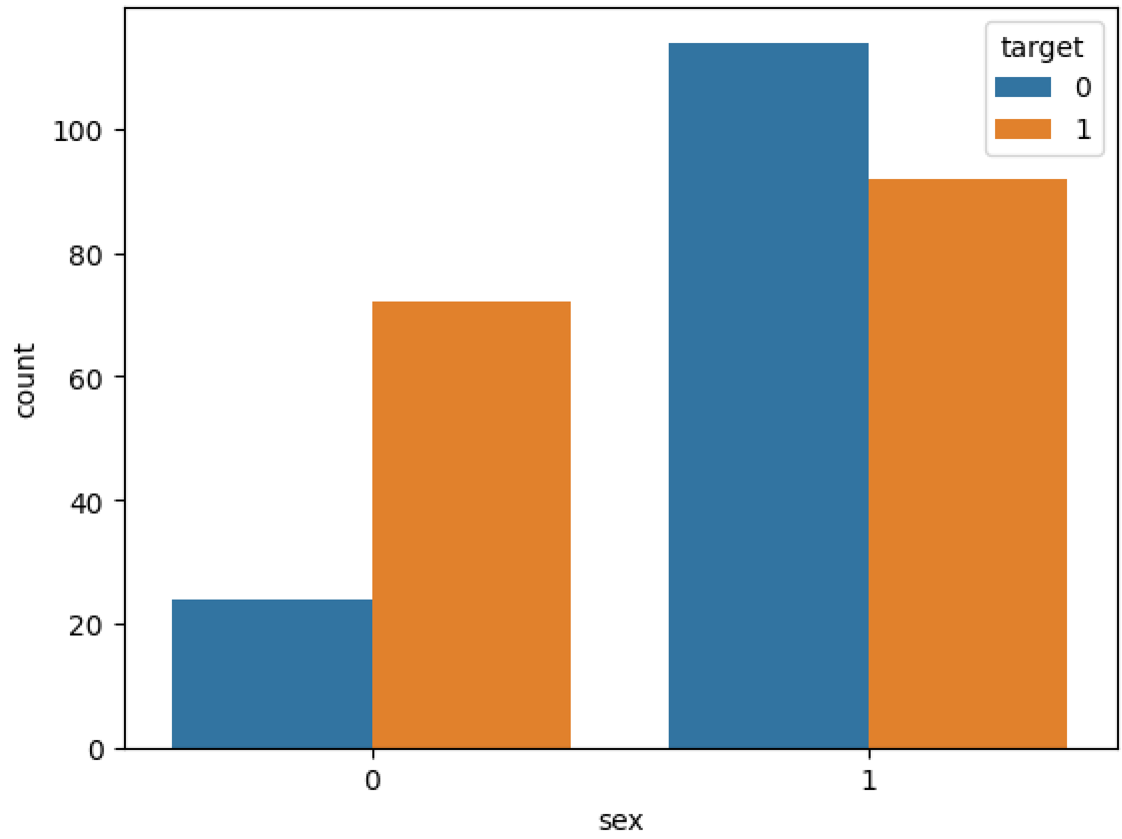
count
160
140
120
100
80
60
40
20
0
0
1
2
3
thal



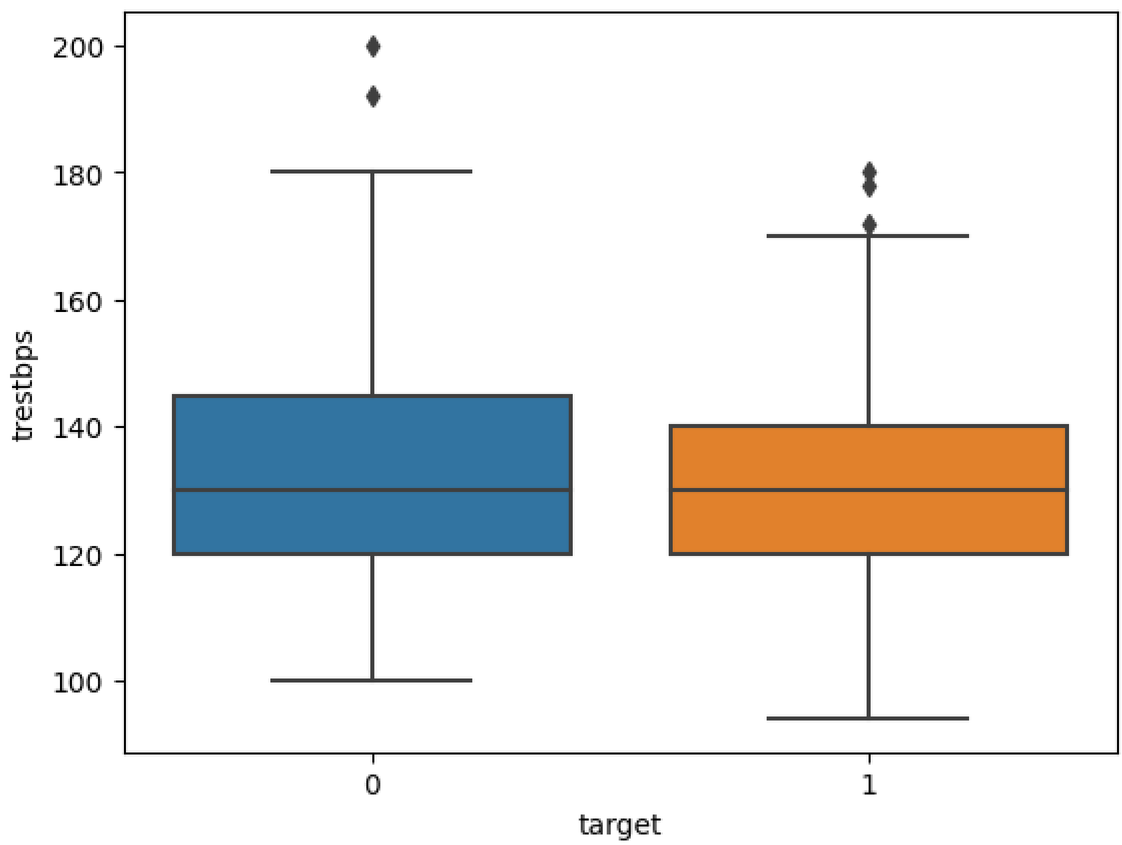
In [18]:
```
# c. Occurrence of CVD across Age category
sns.histplot(x='age', hue='target', data=df, kde=True)
plt.show()
```



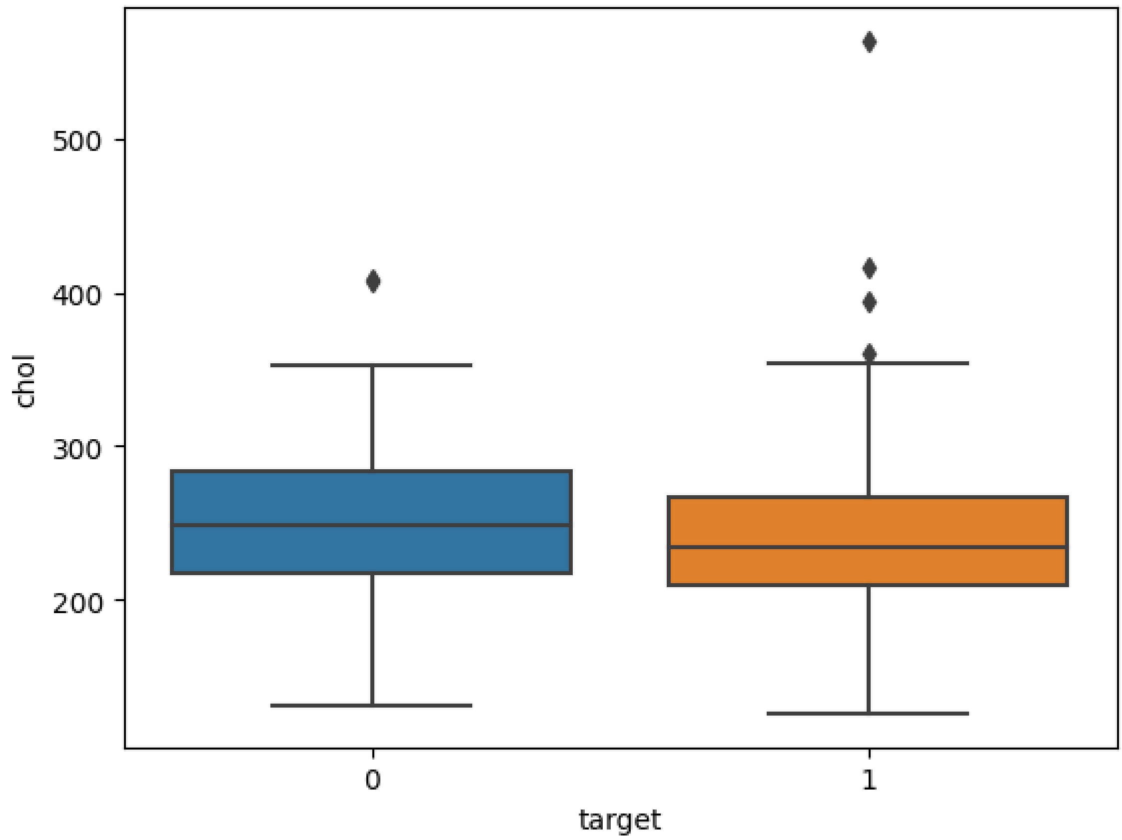
In [20]:
```
# d. Composition of patients with respect to Sex category
sns.countplot(x='sex', hue='target', data=df)
plt.show()
```



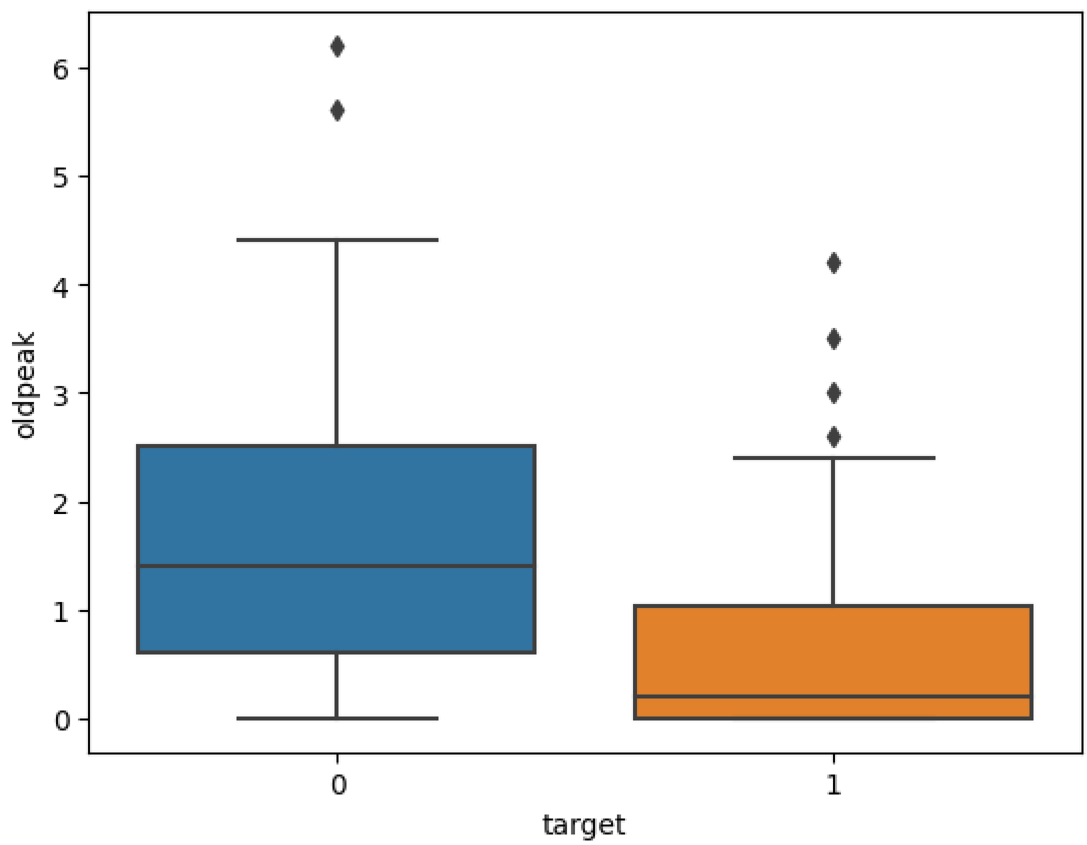
In [22]:
```
# e. Detecting heart attacks based on resting blood pressure anomalies (trestbps
sns.boxplot(x='target', y='trestbps', data=df)
plt.show()
```



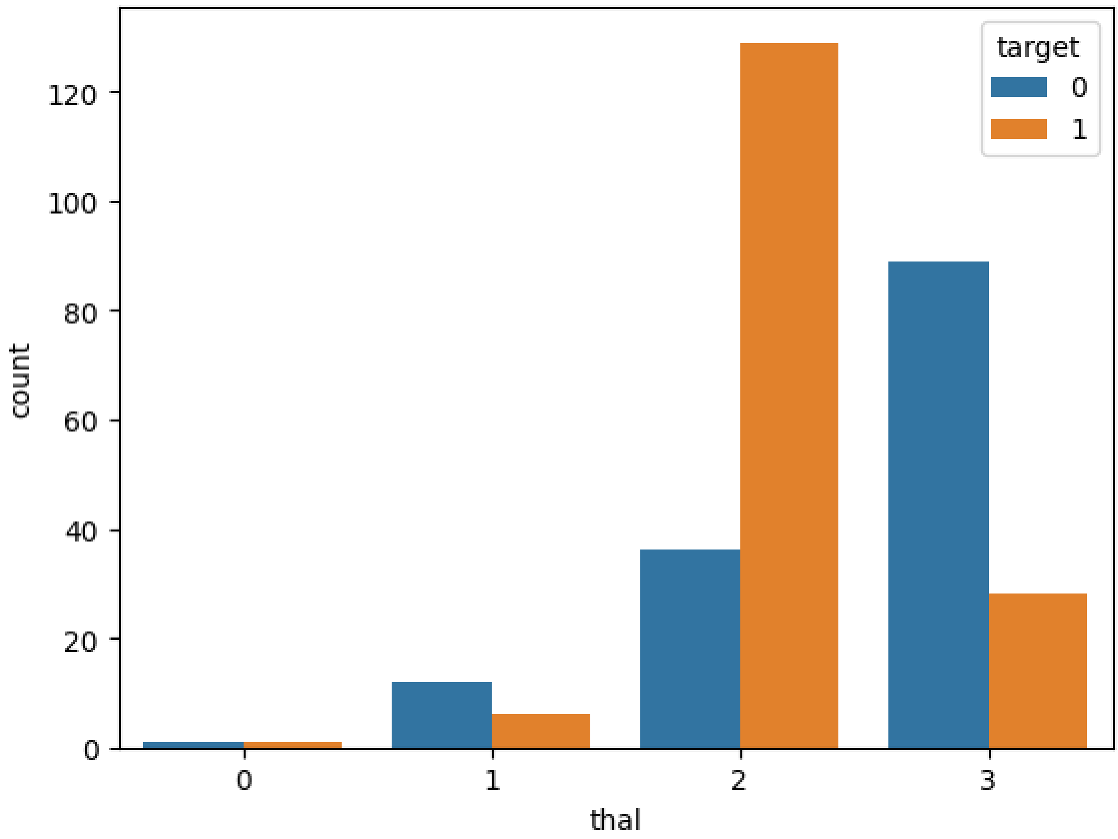
In [24]:
```
# f. Relationship between cholesterol levels and target variable
sns.boxplot(x='target', y='chol', data=df)
plt.show()
```



In [26]:
```
# g. Relationship between peak exercising and the occurrence of a heart attack
sns.boxplot(x='target', y='oldpeak', data=df)
plt.show()
```

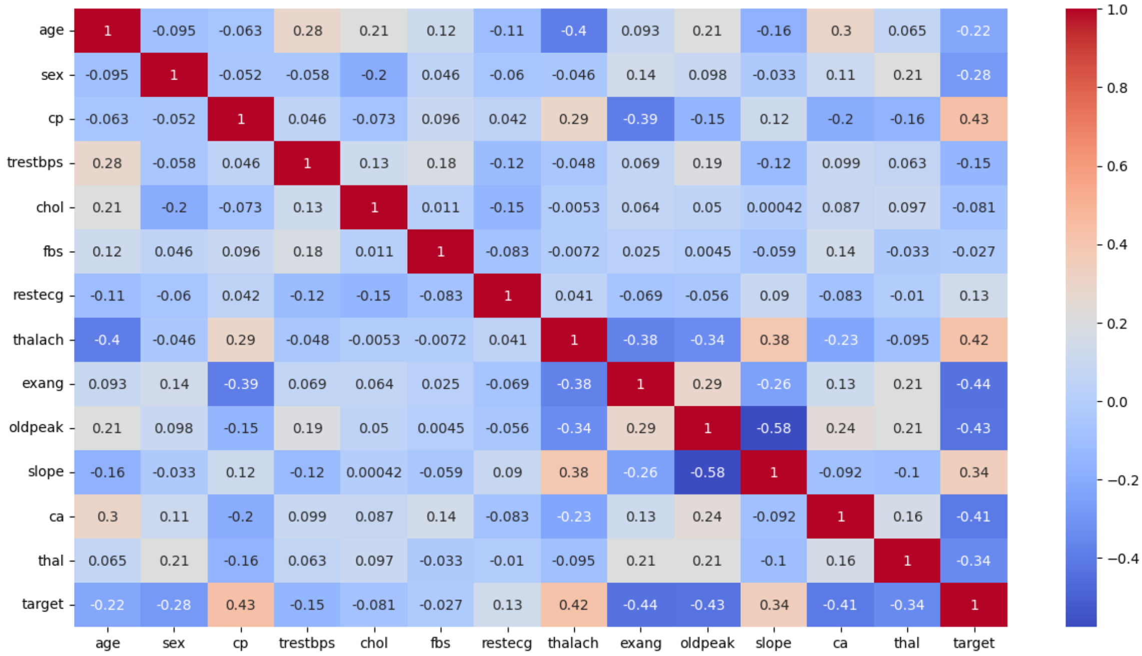


In [27]:
```
# h. Checking if thalassemia is a major cause of CVD
sns.countplot(x='thal', hue='target', data=df)
plt.show()
```



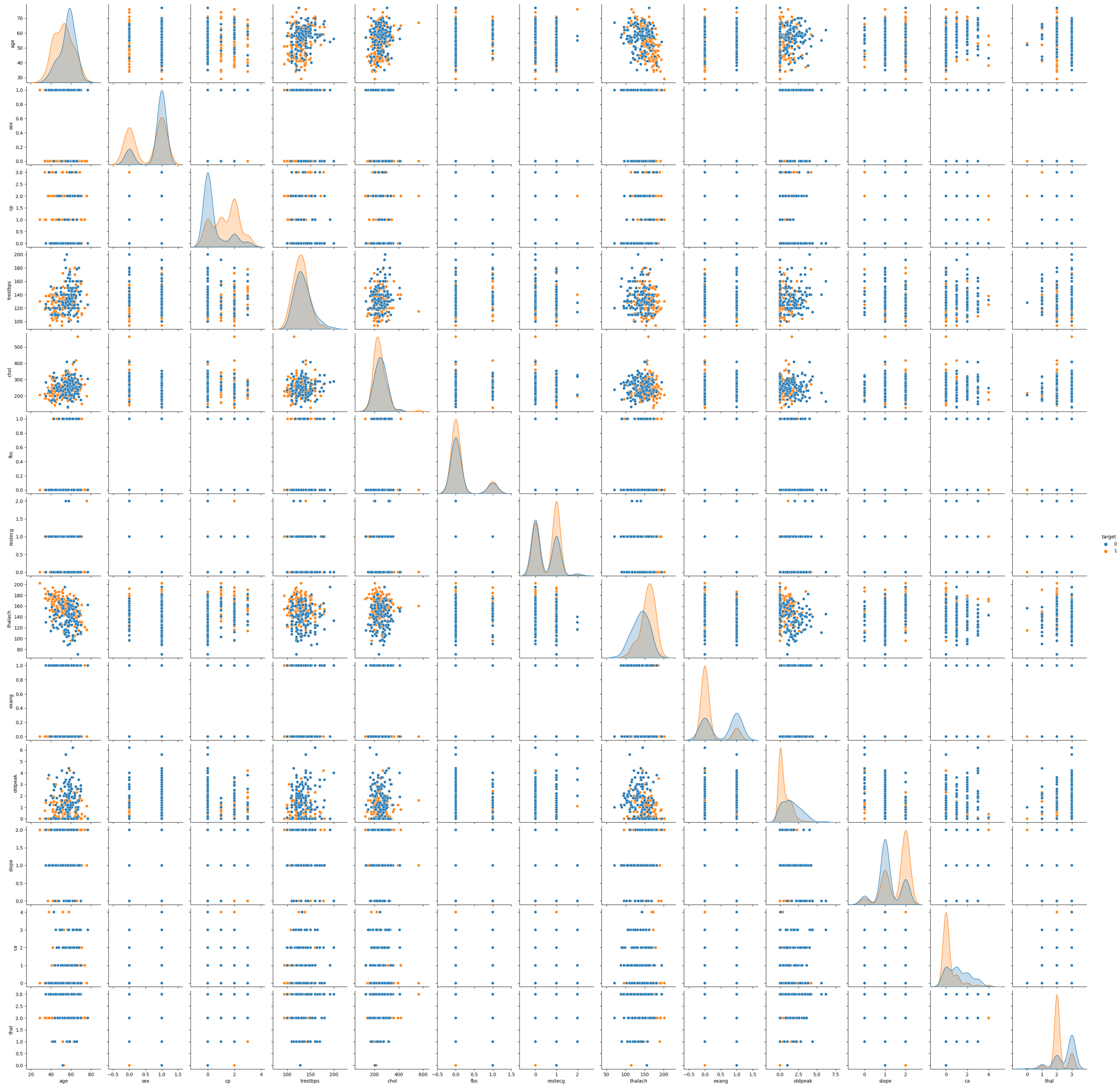
In [32]:

```python
# Define figsize
figsize = (15, 8)

# Factors determining the occurrence of CVD
corr_matrix = df.corr()
plt.figure(figsize=figsize)
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```



In [34]:

```python
# j. Relationship between all given variables
sns.pairplot(df, hue='target')
plt.show()
```



3. Build a baseline model

In [35]:
```
# Select features and target variable
X = df.drop('target', axis=1)
y = df['target']
```

In [36]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [42]:
```
# Logistic Regression model
logreg = LogisticRegression(max_iter = 1000)
logreg.fit(X_train, y_train)
```

Out[42]:
```
▾         LogisticRegression
LogisticRegression(max_iter=1000)
```

In [43]:
```
# Random Forest model
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
```

Out[43]:

```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [44]:

```
# Evaluate the models
print('Logistic Regression:')
print(classification_report(y_test, logreg.predict(X_test)))
print(confusion_matrix(y_test, logreg.predict(X_test)))

print('Random Forest:')
print(classification_report(y_test, rf.predict(X_test)))
print(confusion_matrix(y_test, rf.predict(X_test)))
```

```
Logistic Regression:
              precision    recall  f1-score   support

           0       0.80      0.83      0.81        29
           1       0.84      0.81      0.83        32

    accuracy                           0.82        61
   macro avg       0.82      0.82      0.82        61
weighted avg       0.82      0.82      0.82        61

[[24  5]
 [ 6 26]]
Random Forest:
              precision    recall  f1-score   support

           0       0.81      0.90      0.85        29
           1       0.90      0.81      0.85        32

    accuracy                           0.85        61
   macro avg       0.85      0.85      0.85        61
weighted avg       0.86      0.85      0.85        61

[[26  3]
 [ 6 26]]
```

Thank You