

```
In [177... import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

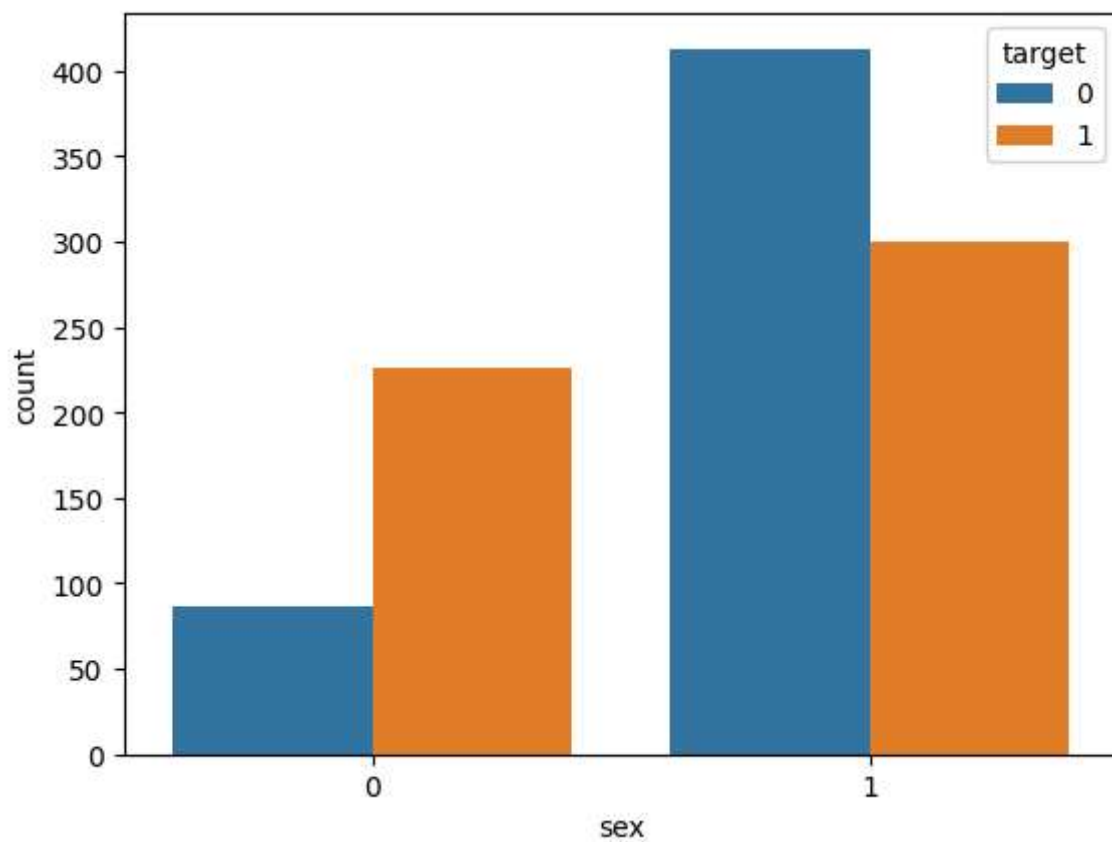
```
In [178... df=pd.read_csv("heart.csv")
df.head()
```

```
Out[178...   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  ta
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | ta |
|---|-----|-----|----|----------|------|-----|---------|---------|-------|---------|-------|----|------|----|
| 0 | 52 | 1 | 0 | 125 | 212 | 0 | 1 | 168 | 0 | 1.0 | 2 | 2 | 3 | |
| 1 | 53 | 1 | 0 | 140 | 203 | 1 | 0 | 155 | 1 | 3.1 | 0 | 0 | 3 | |
| 2 | 70 | 1 | 0 | 145 | 174 | 0 | 1 | 125 | 1 | 2.6 | 0 | 0 | 3 | |
| 3 | 61 | 1 | 0 | 148 | 203 | 0 | 1 | 161 | 0 | 0.0 | 2 | 1 | 3 | |
| 4 | 62 | 0 | 0 | 138 | 294 | 1 | 1 | 106 | 0 | 1.9 | 1 | 3 | 2 | |

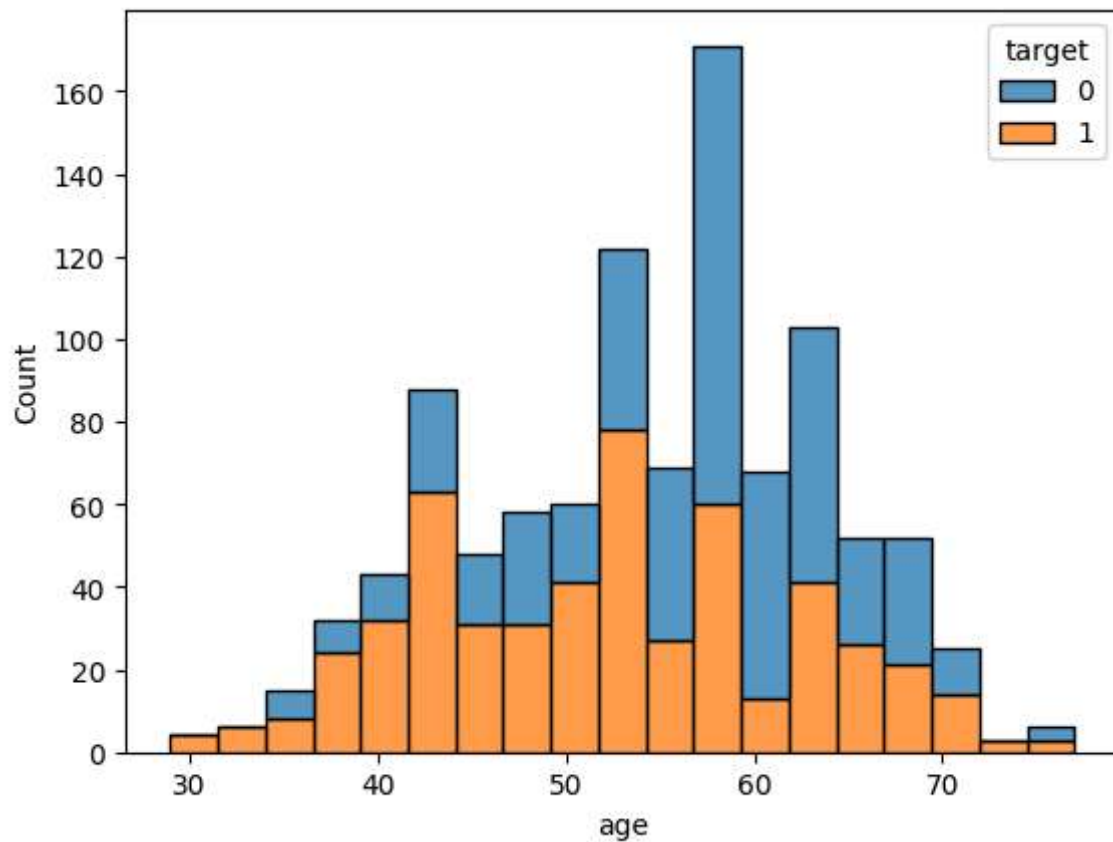
```
In [179... sns.countplot(data=df,x="sex",hue="target")
```

```
Out[179... <Axes: xlabel='sex', ylabel='count'>
```



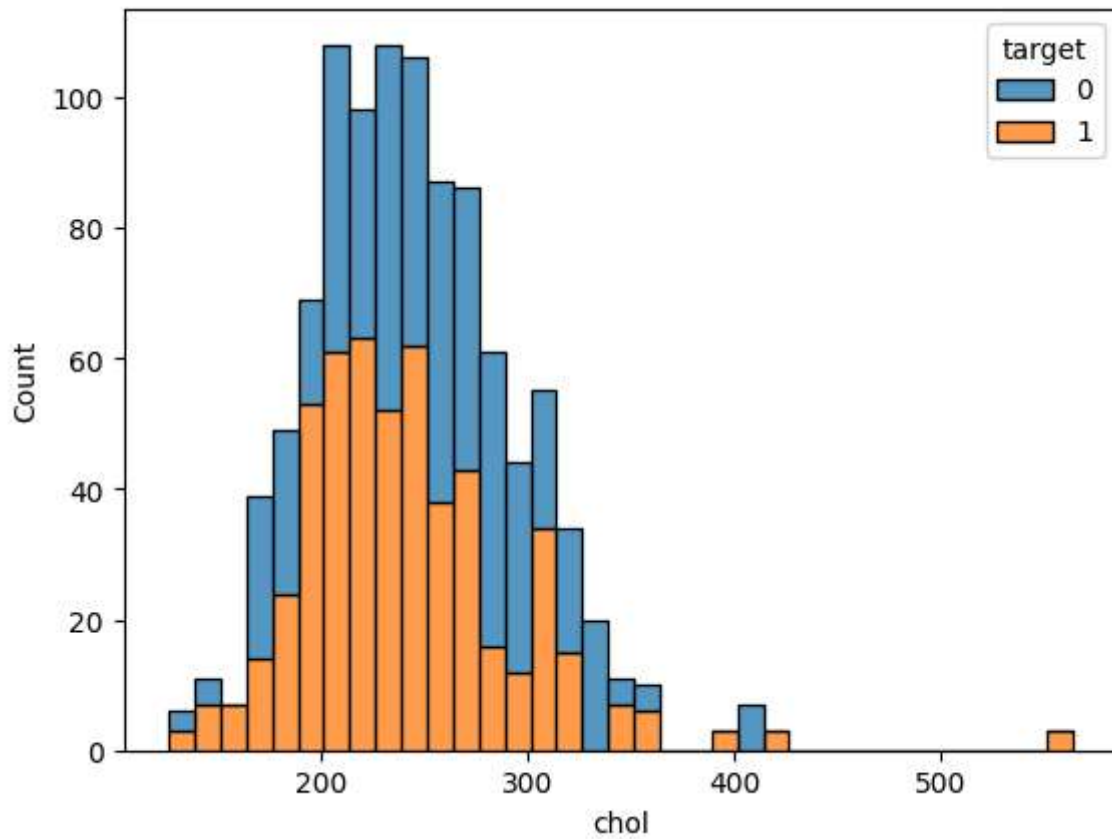
```
In [180... sns.histplot(data=df,x="age",hue="target",multiple="stack")
```

Out[180... <Axes: xlabel='age', ylabel='Count'>



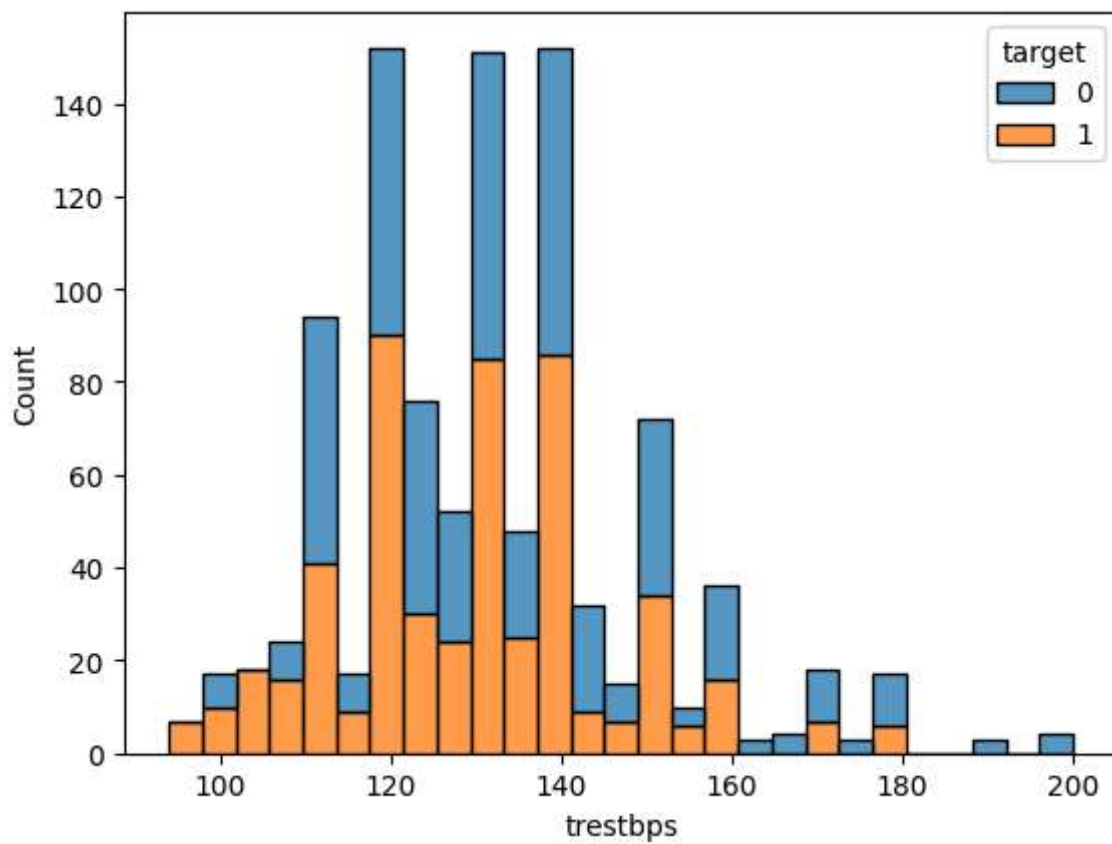
```
In [181... sns.histplot(data=df,x="chol",hue="target",multiple="stack")
```

Out[181... <Axes: xlabel='chol', ylabel='Count'>



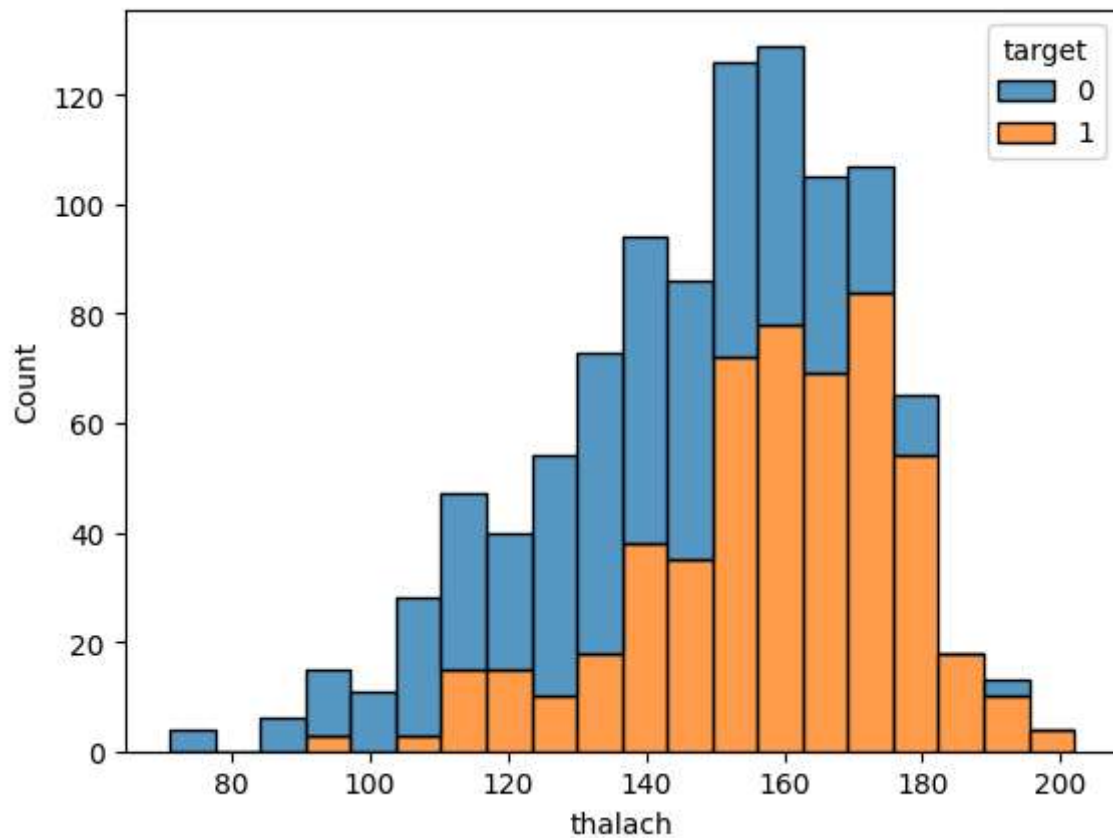
```
In [182...] sns.histplot(data=df,x="trestbps",hue="target",multiple="stack")
```

```
Out[182...] <Axes: xlabel='trestbps', ylabel='Count'>
```



```
In [183...] sns.histplot(data=df,x="thalach",hue="target",multiple="stack")
```

```
Out[183...] <Axes: xlabel='thalach', ylabel='Count'>
```

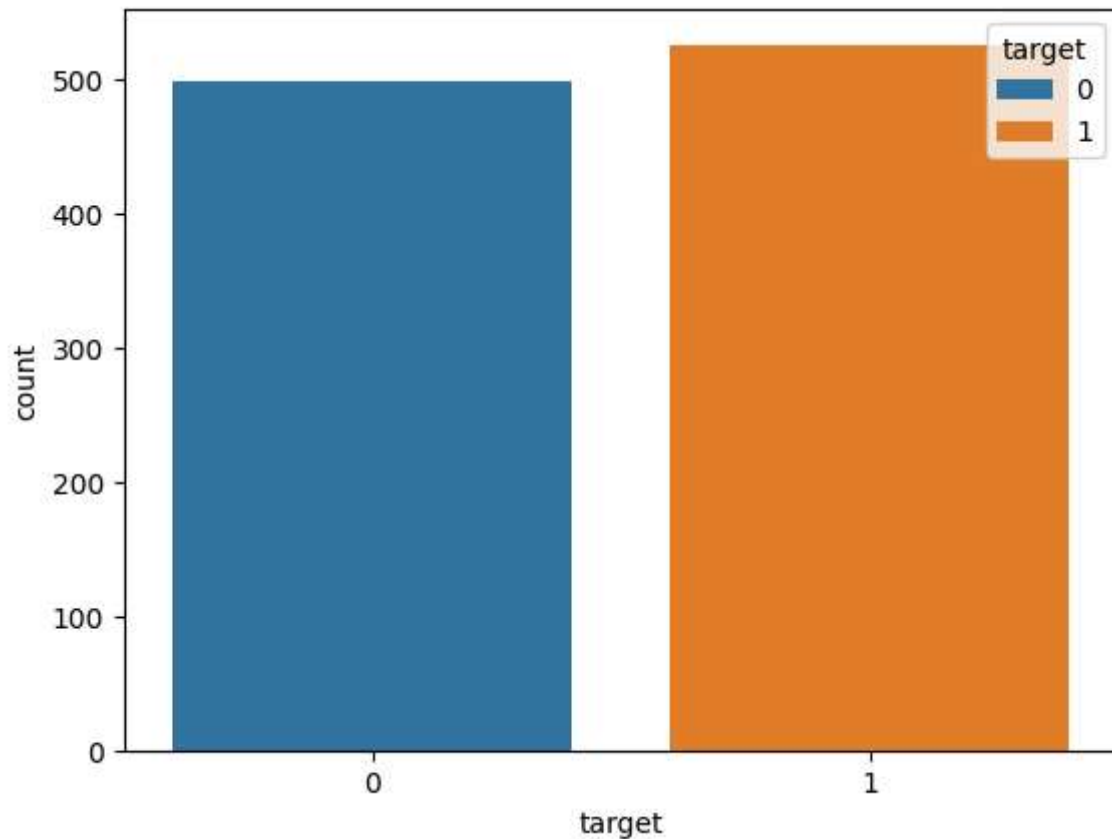


```
In [184...] df.isnull().sum()
```

```
Out[184...] 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 [185...] sns.countplot(data=df,x="target",hue="target")
print (df.target.value_counts())
```

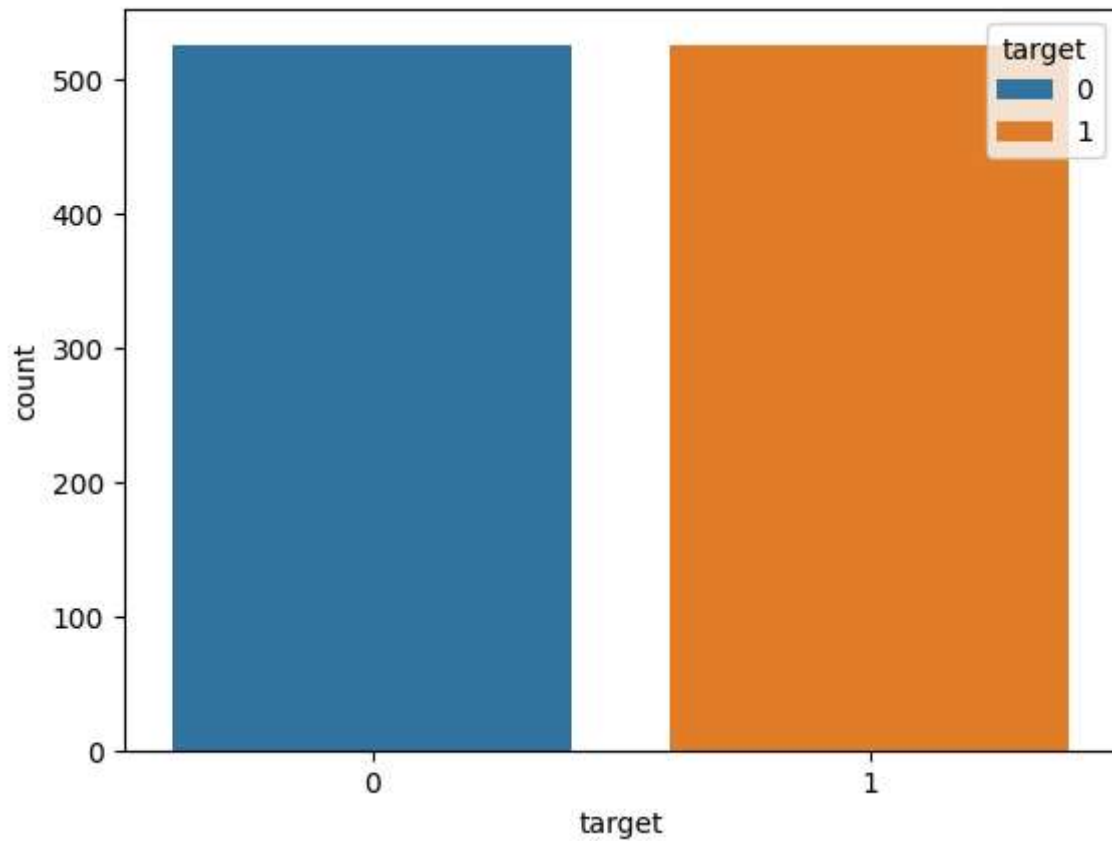
```
target
1    526
0    499
Name: count, dtype: int64
```



```
In [186... from sklearn.utils import resample
df_majority=df[(df['target']==1)]
df_minority=df[(df['target']==0)]
df_minority_upsampled=resample(df_minority,n_samples=526,random_state=0)
df=pd.concat([df_majority,df_minority_upsampled])
```

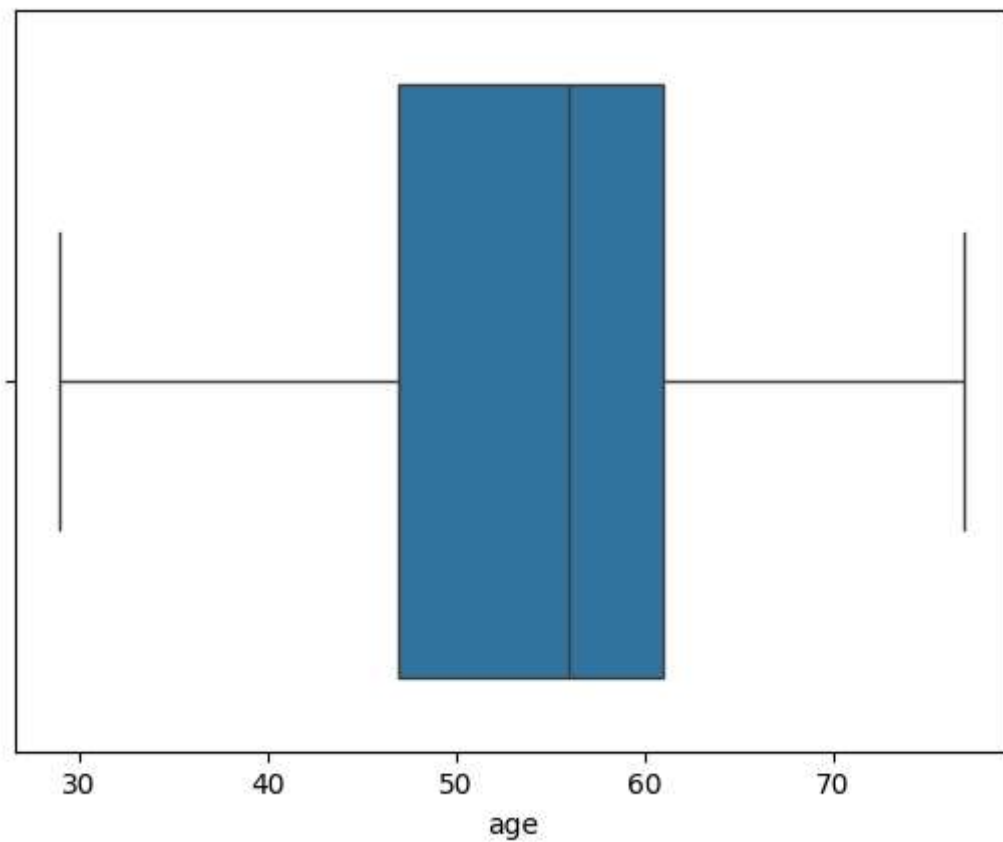
```
In [187... sns.countplot(data=df,x="target",hue="target")
print (df.target.value_counts())
```

```
target
1    526
0    526
Name: count, dtype: int64
```



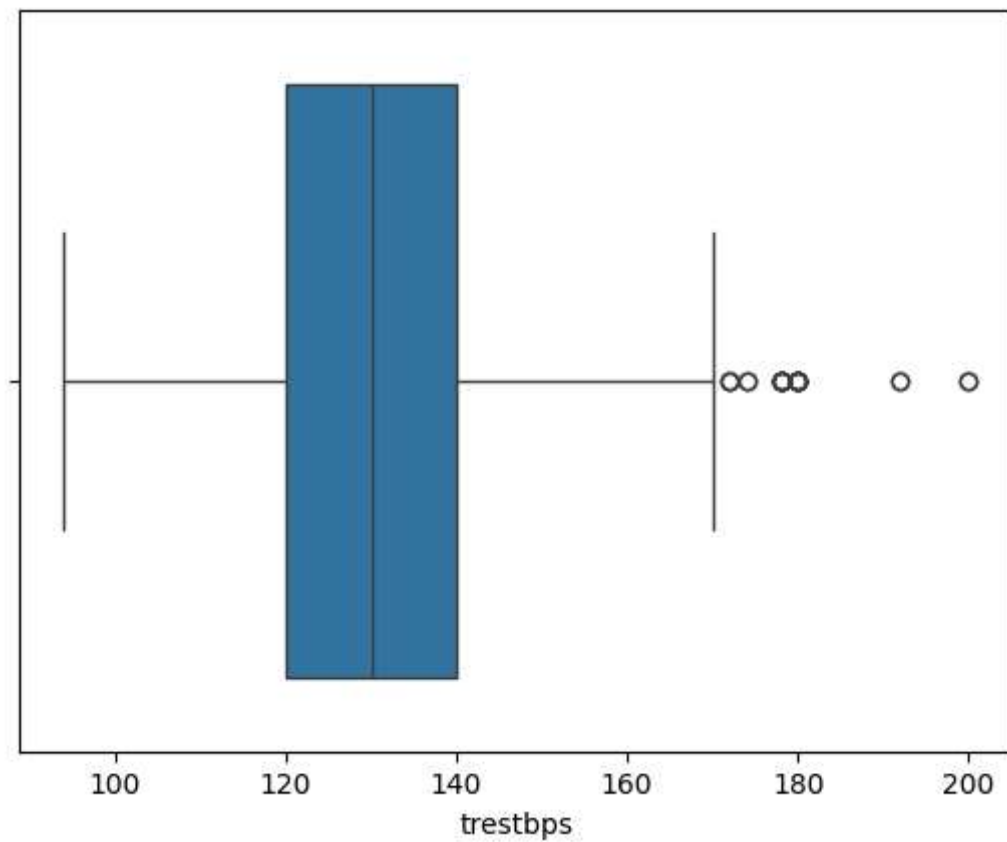
```
In [188... sns.boxplot(data=df, x="age")
```

```
Out[188... <Axes: xlabel='age'>
```



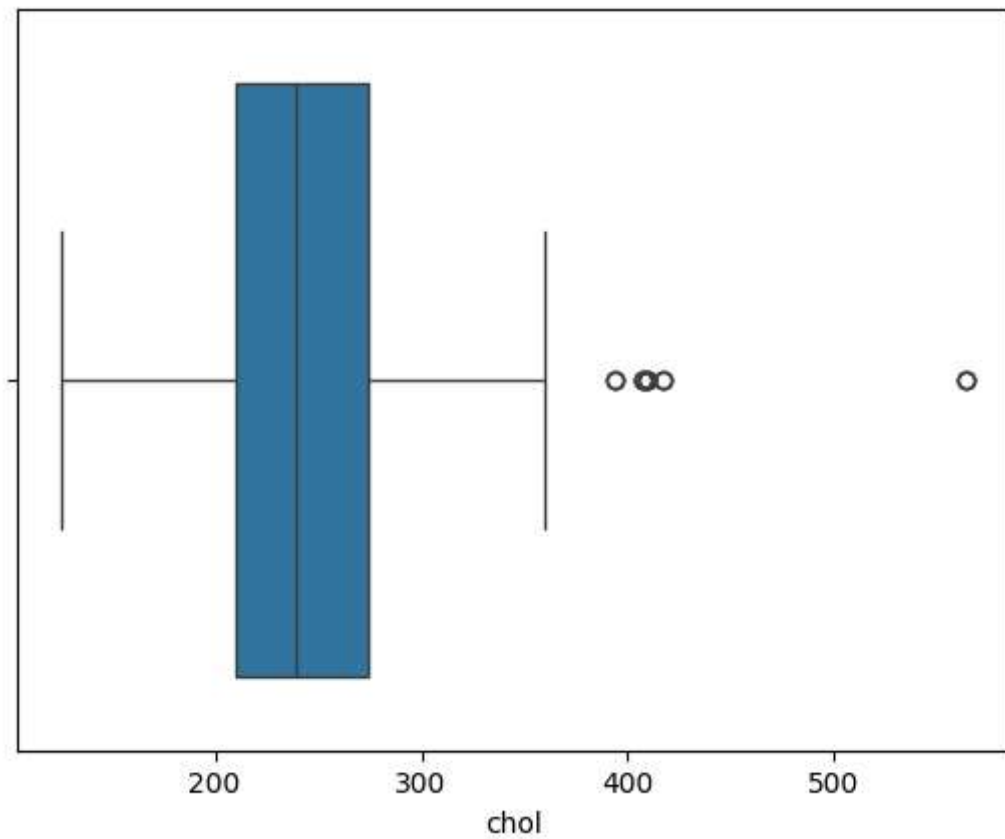
```
In [189... sns.boxplot(data=df,x="trestbps")
```

```
Out[189... <Axes: xlabel='trestbps'>
```



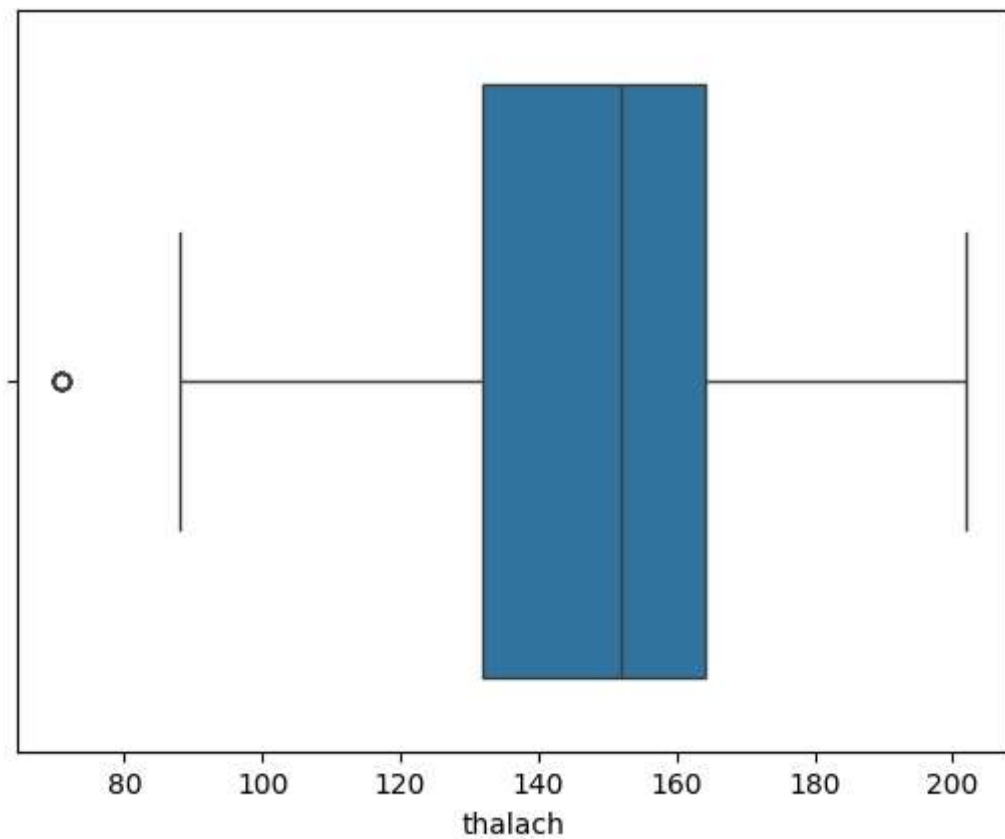
```
In [190... sns.boxplot(data=df,x="chol")
```

```
Out[190... <Axes: xlabel='chol'>
```



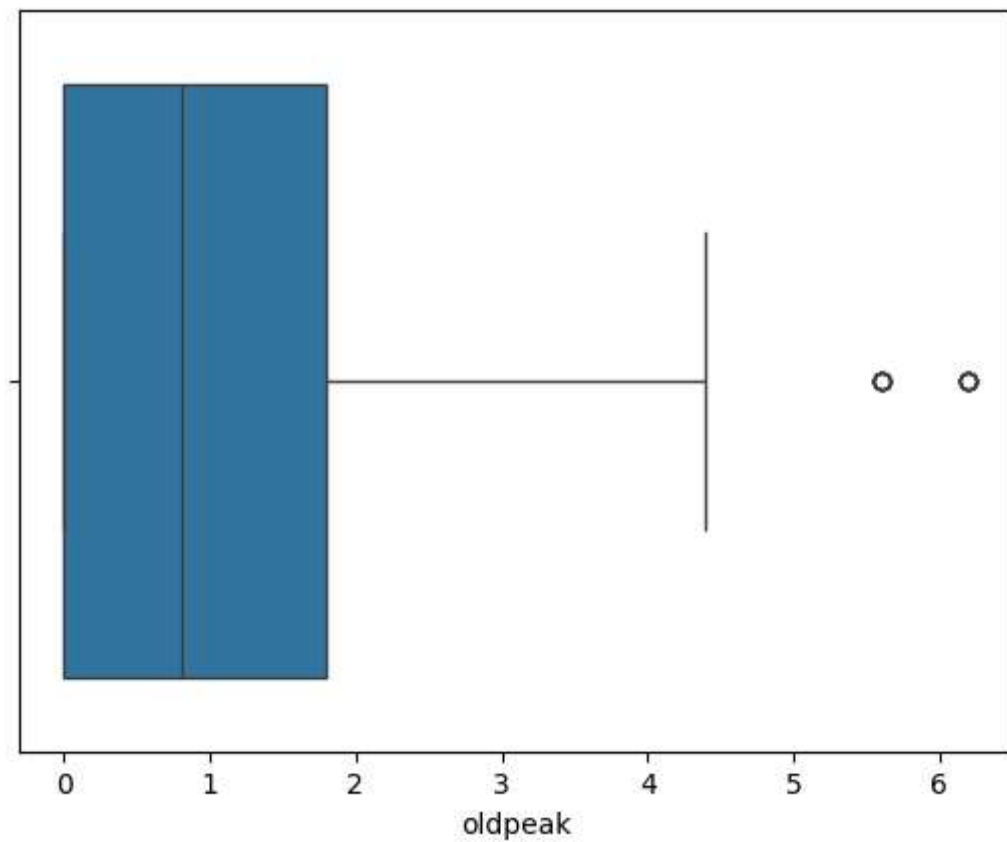
```
In [191... sns.boxplot(data=df,x="thalach")
```

```
Out[191... <Axes: xlabel='thalach'>
```



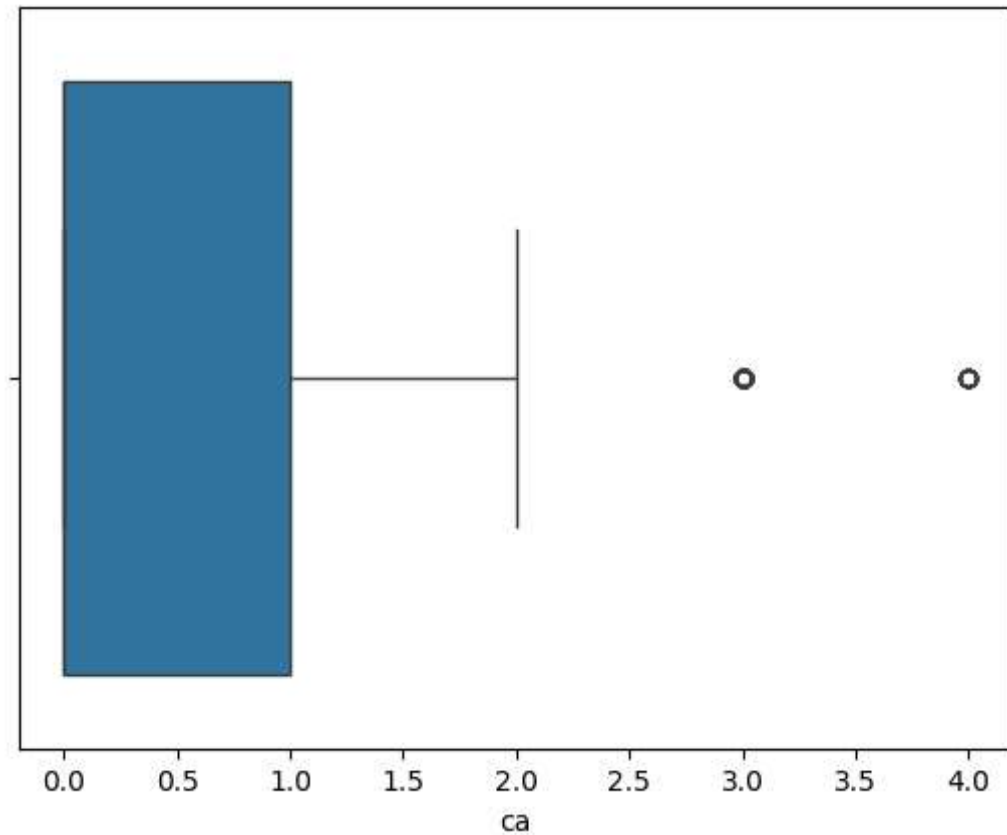

```
In [192... sns.boxplot(data=df,x="oldpeak")
```

```
Out[192... <Axes: xlabel='oldpeak'>
```



```
In [193... sns.boxplot(data=df,x="ca")
```

```
Out[193... <Axes: xlabel='ca'>
```

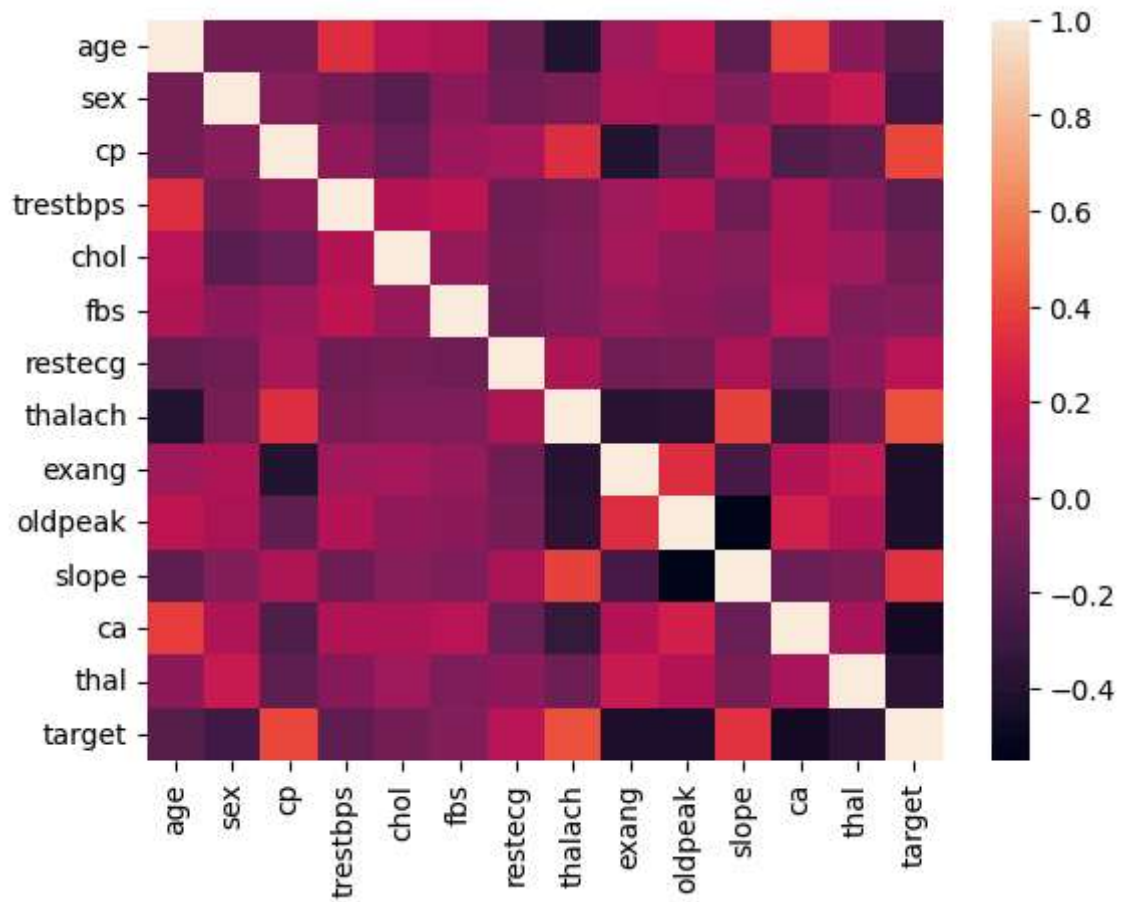


```
In [194... import scipy.stats as stats
z = np.abs(stats.zscore(df))
data_clean = df[(z<3).all(axis = 1)]
data_clean.shape
```

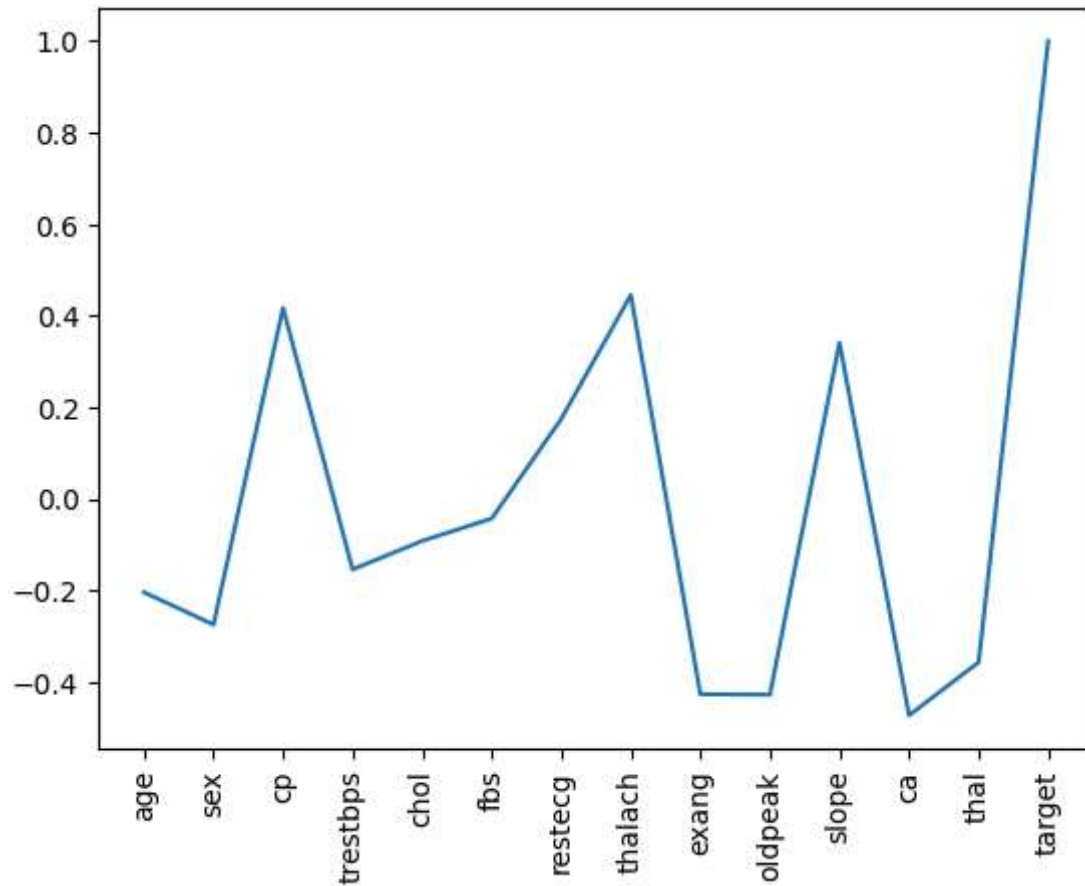
```
Out[194... (989, 14)
```

```
In [195... sns.heatmap(data_clean.corr())
```

```
Out[195... <Axes: >
```



```
In [196... corr = data_clean.corr()['target']  
plt.plot(corr)  
plt.xticks(rotation=90)  
plt.show()
```



```
In [197... x=data_clean.drop("target",axis=1)
y=data_clean["target"]
```

```
In [198... from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,f1_score,confusion_matrix,classification
```

```
In [199... x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

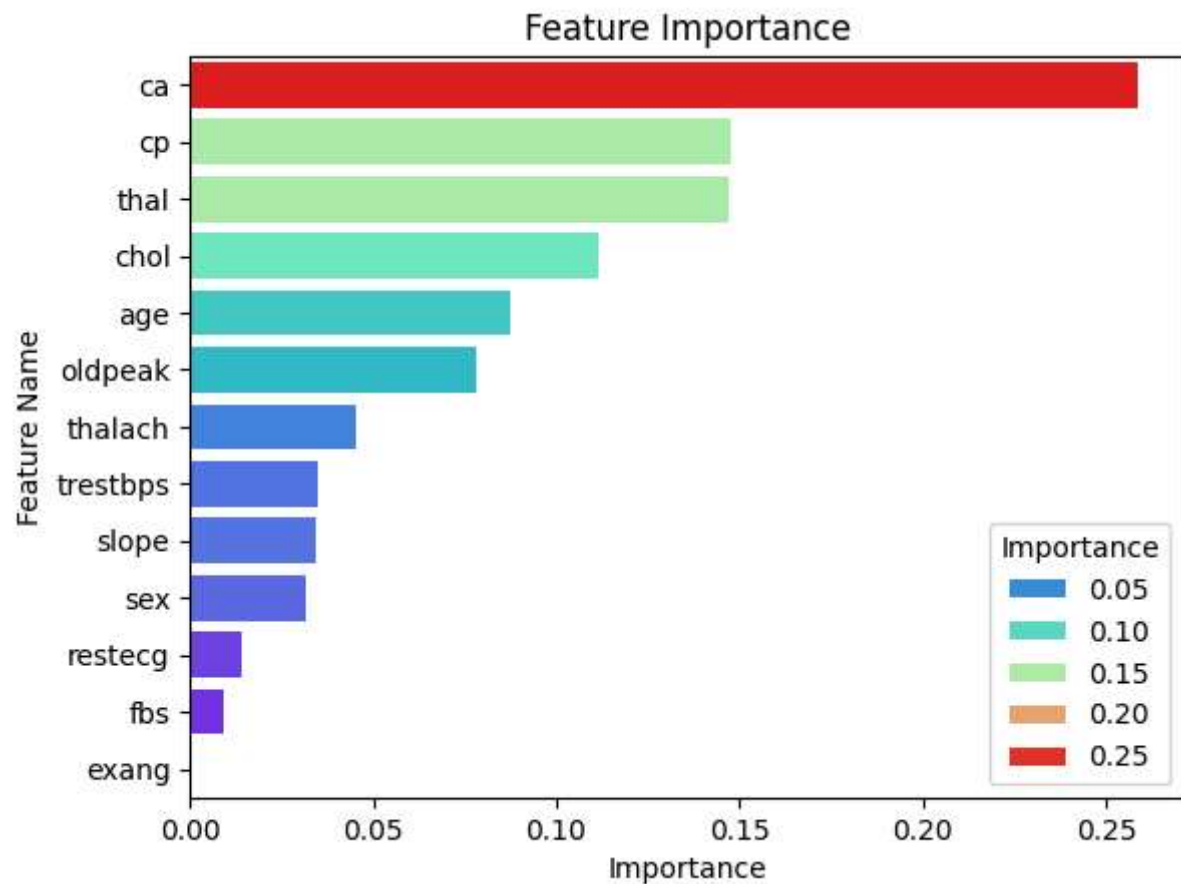
```
In [200... def model_fit(model):
    model.fit(x_train,y_train)
    y_pred=model.predict(x_test)
    model_acc=round(accuracy_score(y_test,y_pred)*100,2)
    print(f"The accuracy of the model is: {model_acc}%")
    print("")
    print("f1_score: ",f1_score(y_test,y_pred))
    print("precision_score: ",precision_score(y_test,y_pred))
    print("recall_score: ",recall_score(y_test,y_pred))
    print("")
    imp_df=pd.DataFrame({
        "Feature Name":x_train.columns,
        "Importance":model.feature_importances_
    })
    imp_df=imp_df.sort_values(by="Importance",ascending=False)
    sns.barplot(data=imp_df,x="Importance",y="Feature Name",hue="Importance",palett
plt.title("Feature Importance")
plt.xlabel("Importance")
```

```
plt.ylabel("Feature Name")
```

```
In [201... from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(random_state=0)
model_fit(classifier)
```

The accuracy of the model is: 100.0%

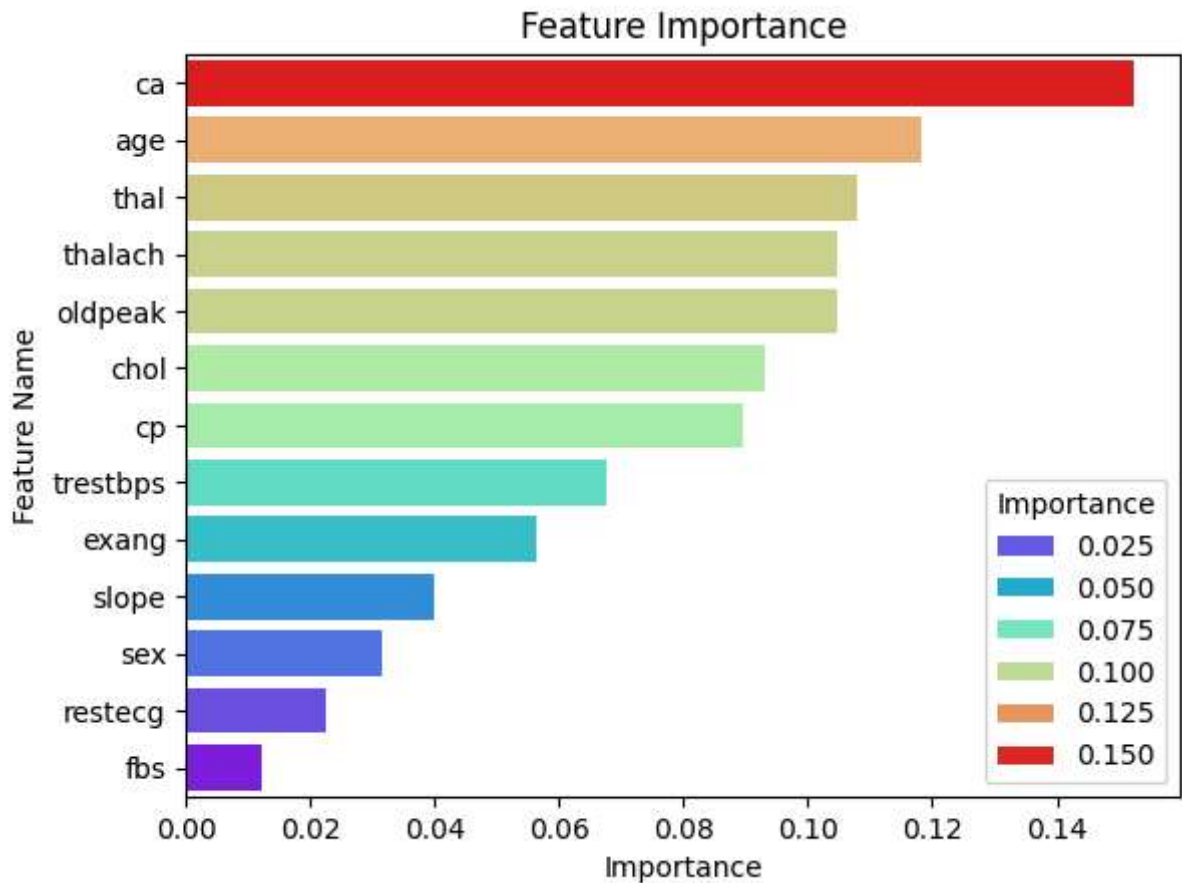
```
f1_score: 1.0
precision_score: 1.0
recall_score: 1.0
```



```
In [202... from sklearn.ensemble import RandomForestClassifier
classifier= RandomForestClassifier(n_estimators= 10, criterion="entropy")
model_fit(classifier)
```

The accuracy of the model is: 99.49%

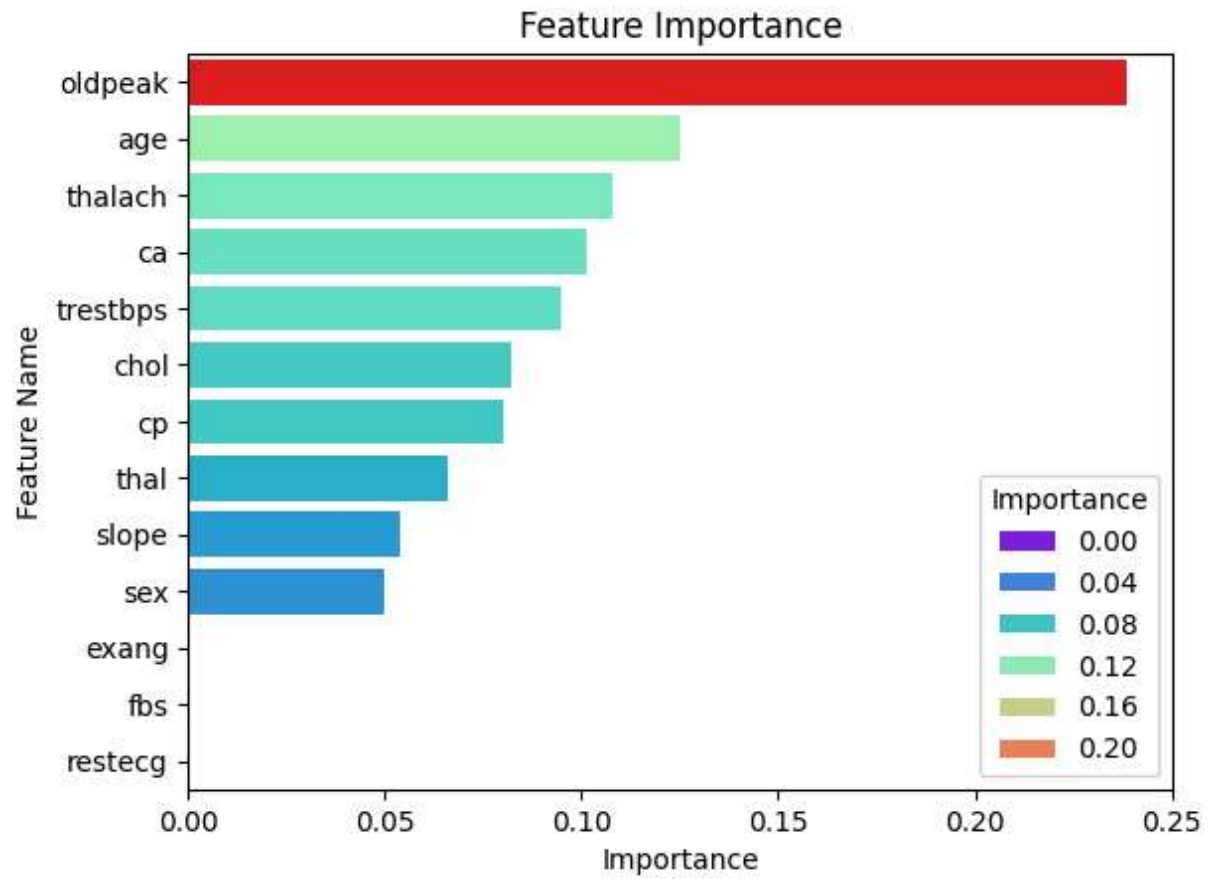
```
f1_score: 0.9945945945945946
precision_score: 0.989247311827957
recall_score: 1.0
```



```
In [203... from sklearn.ensemble import AdaBoostClassifier  
abc = AdaBoostClassifier(n_estimators=50, learning_rate=1)  
model_fit(abc)
```

The accuracy of the model is: 88.38%

f1_score: 0.8770053475935828
precision_score: 0.8631578947368421
recall_score: 0.8913043478260869



In []: