```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
df=pd.read_csv("heart.csv")
```

```
df.head(5)
```

```
   Age Sex ChestPainType  RestingBP  Cholesterol  FastingBS RestingECG  
MaxHR  \
0   40   M           ATA        140          289          0     Normal  
172
1   49   F           NAP        160          180          0     Normal  
156
2   37   M           ATA        130          283          0         ST  
98
3   48   F           ASY        138          214          0     Normal  
108
4   54   M           NAP        150          195          0     Normal  
122

  ExerciseAngina  Oldpeak ST_Slope  HeartDisease  
0              N      0.0       Up             0  
1              N      1.0     Flat             1  
2              N      0.0       Up             0  
3              Y      1.5     Flat             1  
4              N      0.0       Up             0  
```

```
df.shape
```

```
(918, 12)
```

```
df.isna().sum()
```

```
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             918 non-null    int64
 1   Sex             918 non-null    object
 2   ChestPainType   918 non-null    object
 3   RestingBP       918 non-null    int64
 4   Cholesterol     918 non-null    int64
 5   FastingBS       918 non-null    int64
 6   RestingECG      918 non-null    object
 7   MaxHR           918 non-null    int64
 8   ExerciseAngina  918 non-null    object
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object
 11  HeartDisease    918 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

```
df.describe()
```

```
              Age   RestingBP  Cholesterol   FastingBS       MaxHR  \
count  918.000000  918.000000   918.000000  918.000000  918.000000
mean    53.510893  132.396514   198.799564    0.233115  136.809368
std      9.432617   18.514154   109.384145    0.423046   25.460334
min     28.000000    0.000000     0.000000    0.000000   60.000000
25%     47.000000  120.000000   173.250000    0.000000  120.000000
50%     54.000000  130.000000   223.000000    0.000000  138.000000
75%     60.000000  140.000000   267.000000    0.000000  156.000000
max     77.000000  200.000000   603.000000    1.000000  202.000000

          Oldpeak  HeartDisease
count  918.000000    918.000000
mean     0.887364      0.553377
std      1.066570      0.497414
min     -2.600000      0.000000
25%      0.000000      0.000000
50%      0.600000      1.000000
75%      1.500000      1.000000
max      6.200000      1.000000
```

```
df['Sex'].unique()
```

```
array(['M', 'F'], dtype=object)
```

```
df['ChestPainType'].unique()
```

```
array(['ATA', 'NAP', 'ASY', 'TA'], dtype=object)
```

```
df['RestingECG'].unique()
```

```
array(['Normal', 'ST', 'LVH'], dtype=object)
```

```
df['ExerciseAngina'].unique()
```

```
array(['N', 'Y'], dtype=object)
```

```
df['ST_Slope'].unique()
```

```
array(['Up', 'Flat', 'Down'], dtype=object)
```

```
df['HeartDisease'].unique()
```

```
array([0, 1])
```

```
df['FastingBS'].unique()
```

```
array([0, 1])
```

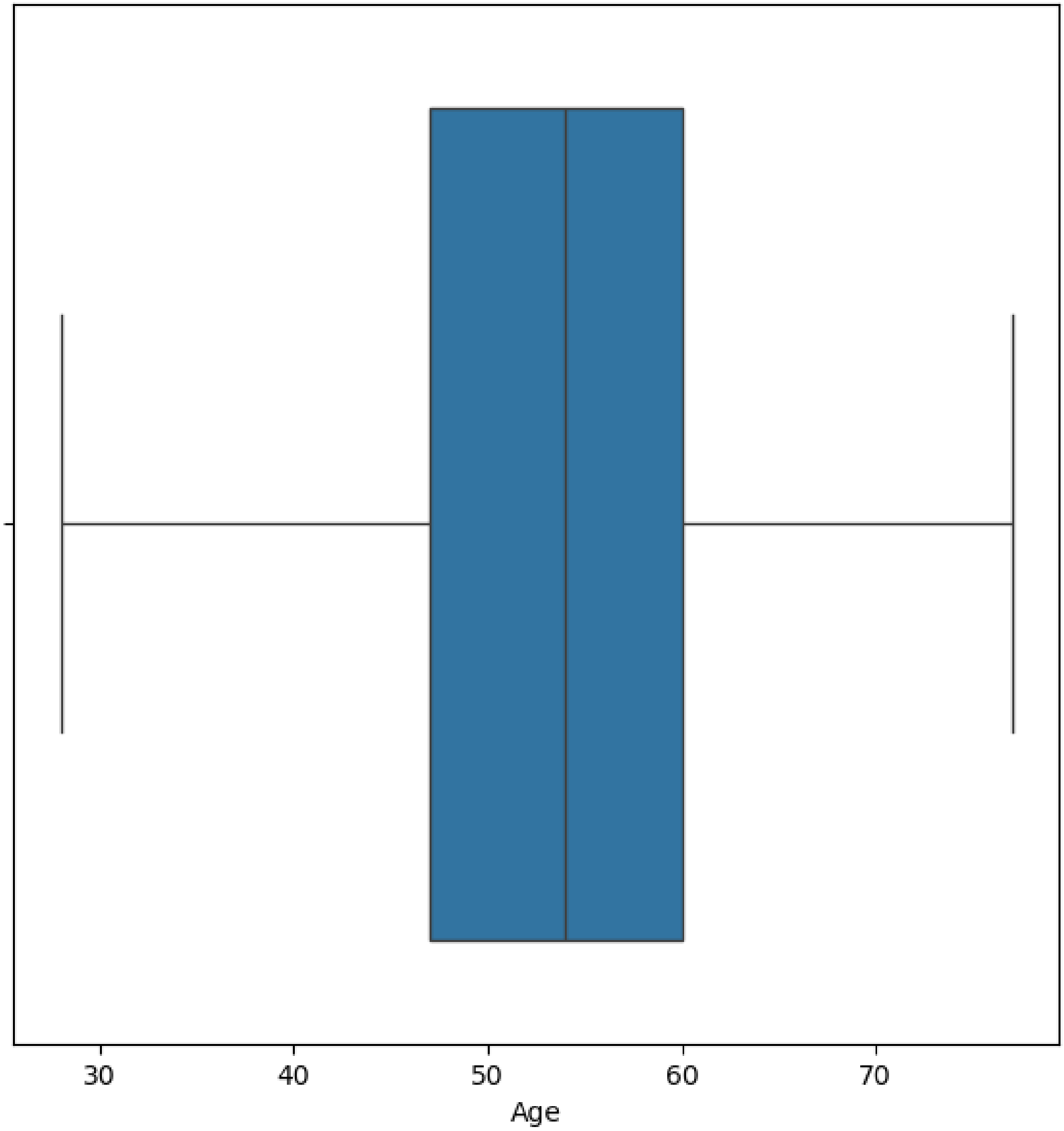
```
df.duplicated().any()
```

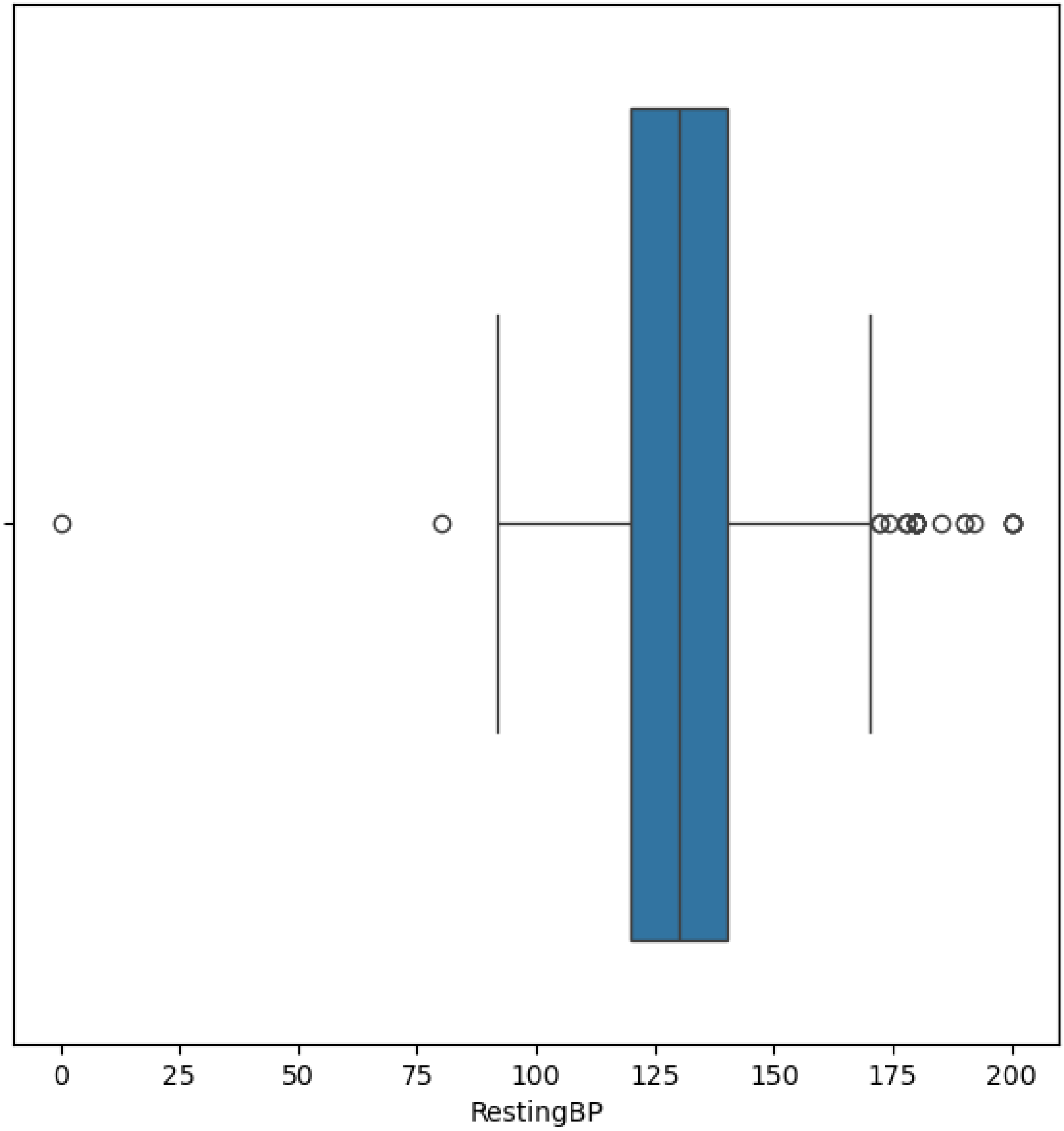
```
np.False_
```

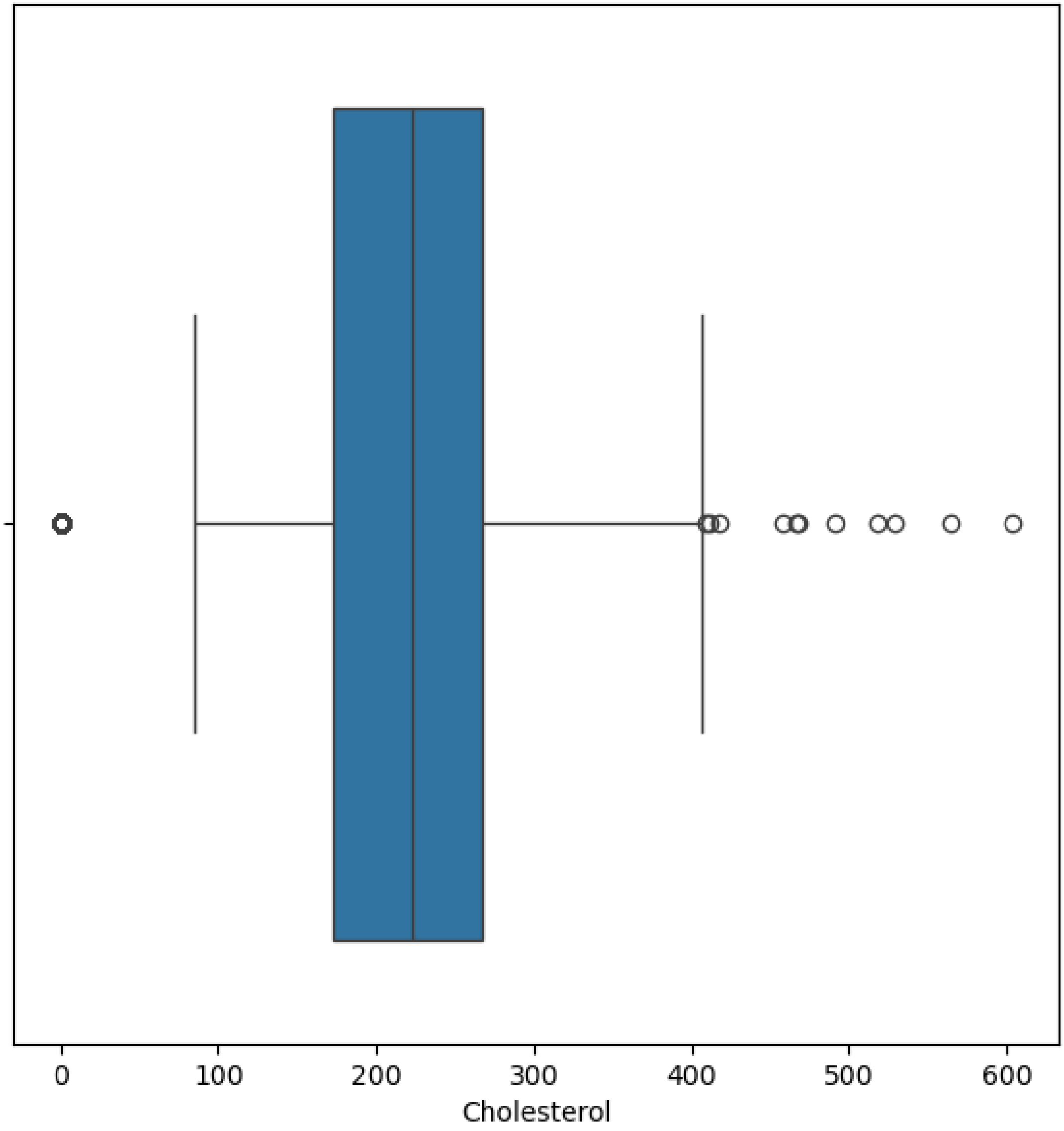
```
df.columns
```

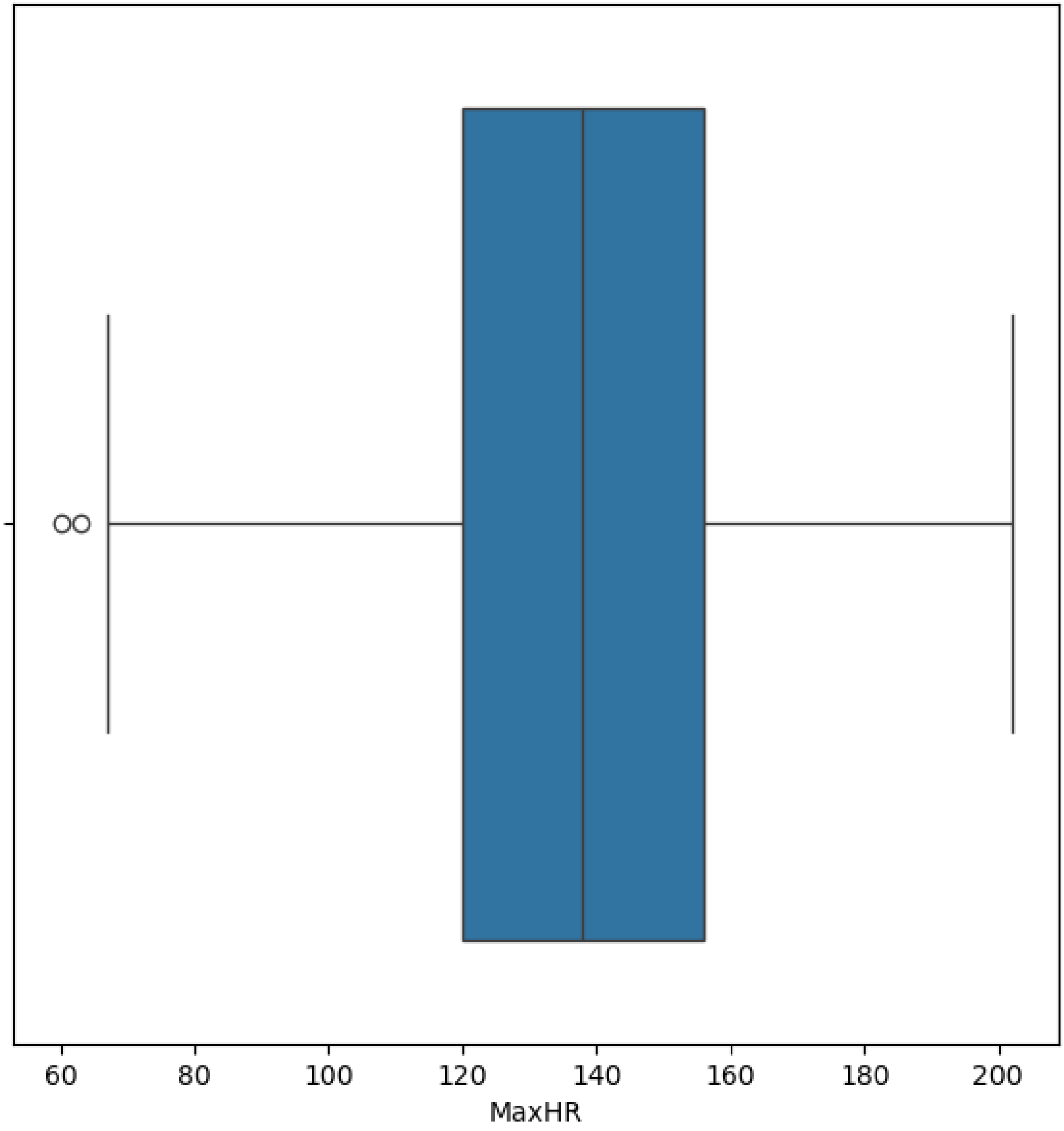
```
Index(['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol',
'FastingBS',
       'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope',
       'HeartDisease'],
      dtype='object')
```

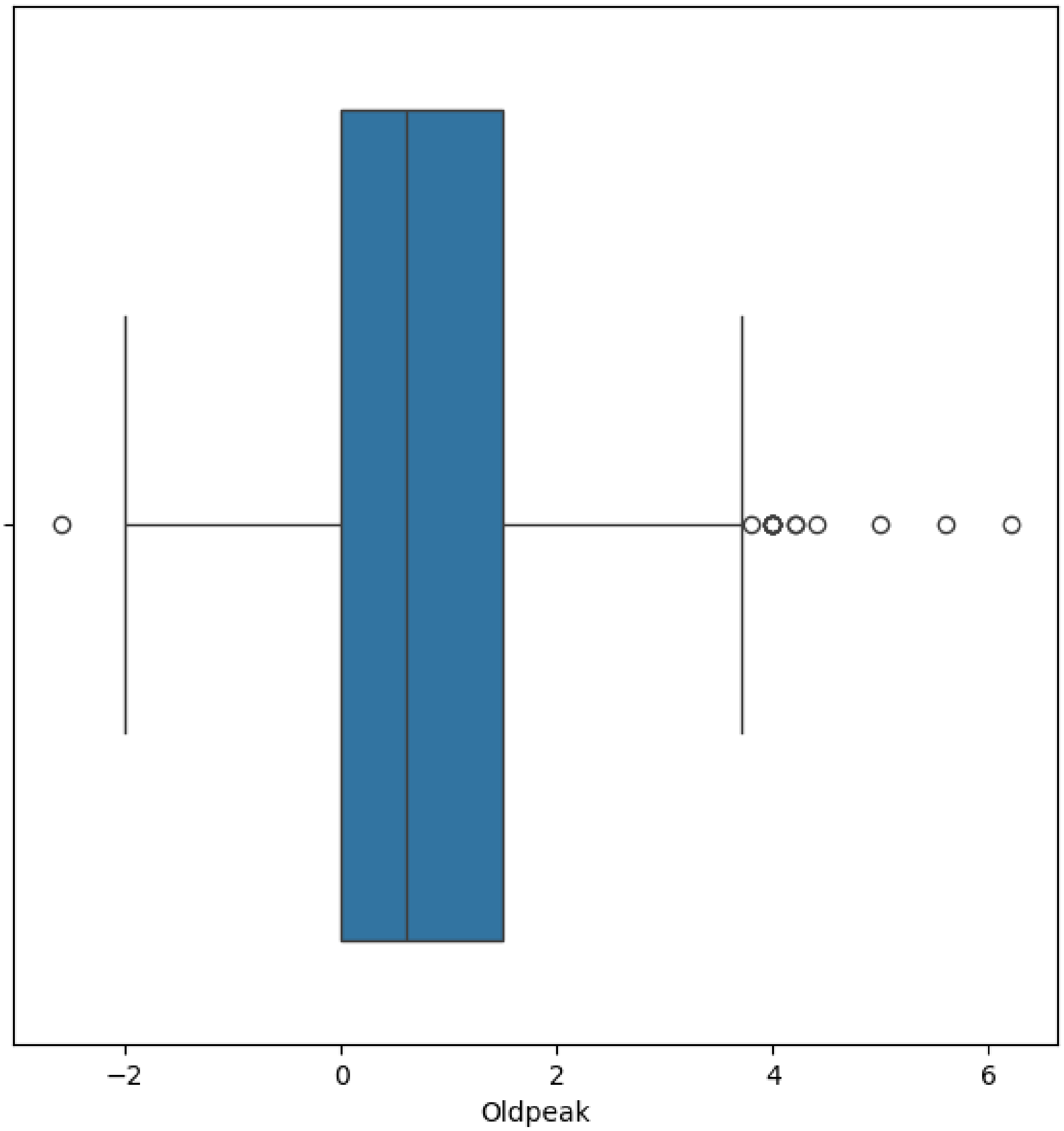
```
cols=['Age', 'RestingBP', 'Cholesterol',
        'MaxHR', 'Oldpeak',
       ]

for i in cols:
    plt.figure(figsize=(7,7))
    sns.boxplot(x=df[i])
```


30
40
50
60
70
Age

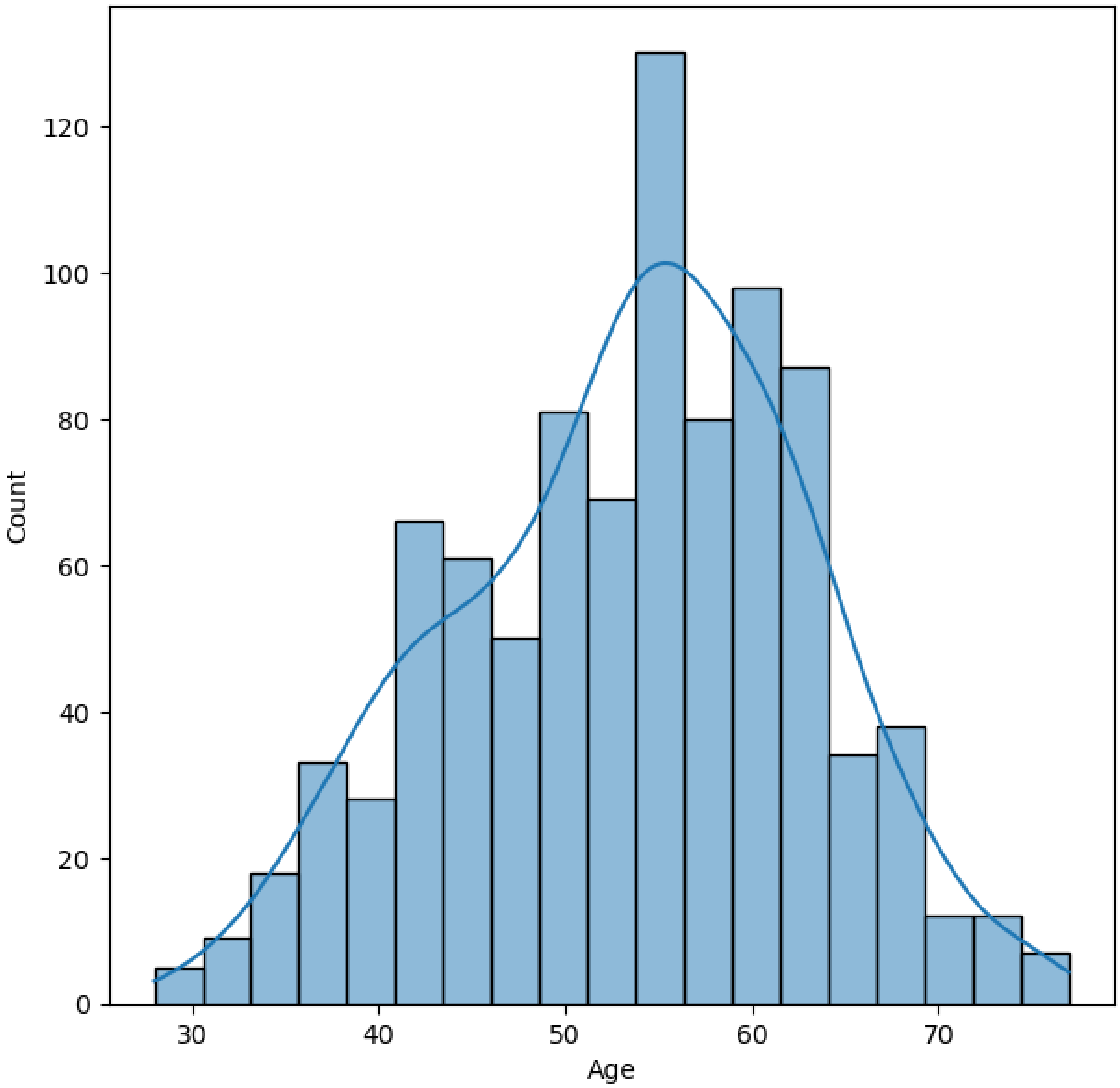

0
25
50
75
100
125
150
175
200
RestingBP

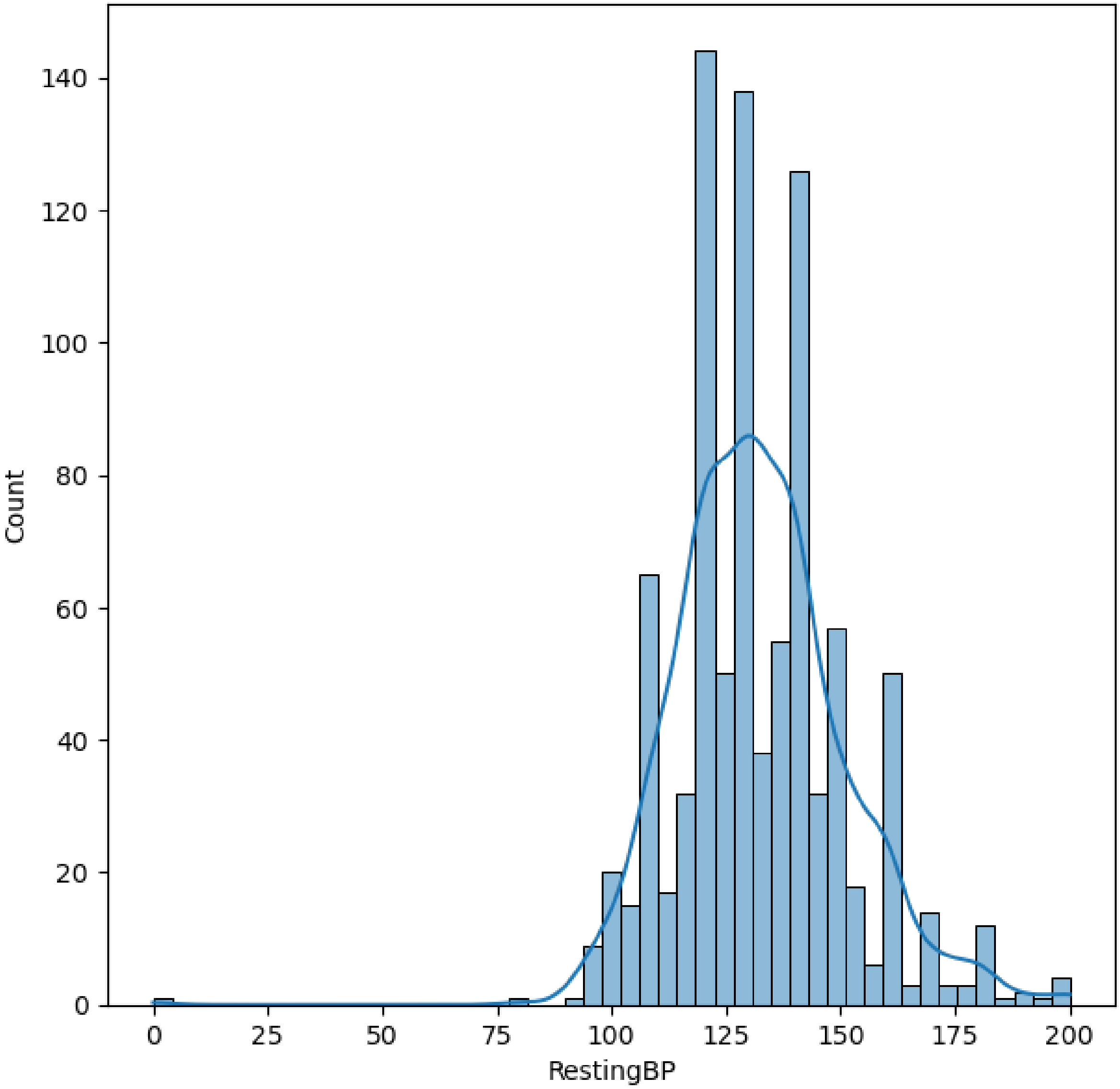

0
100
200
300
400
500
600
Cholesterol

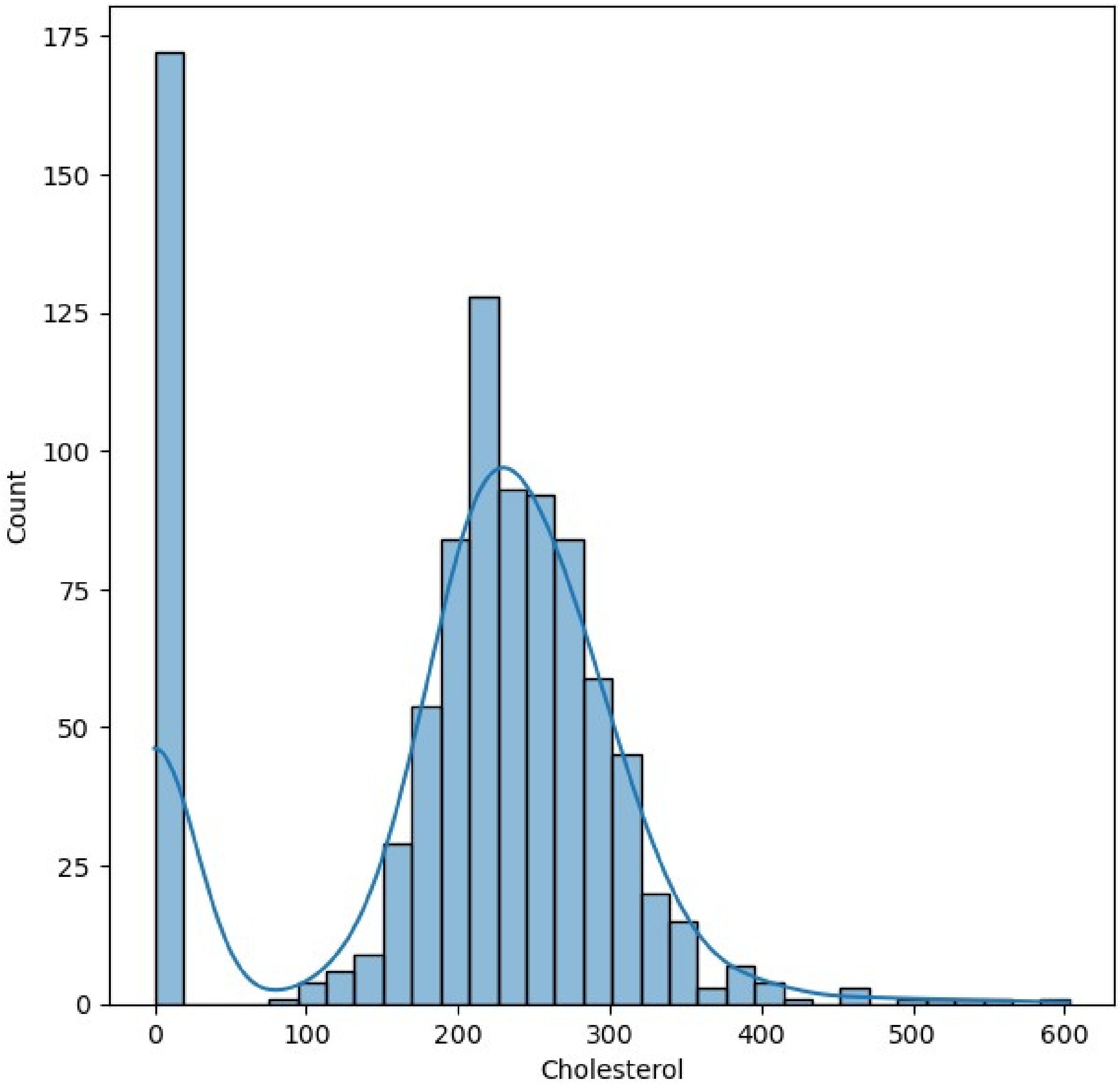

60
80
100
120
140
160
180
200
MaxHR

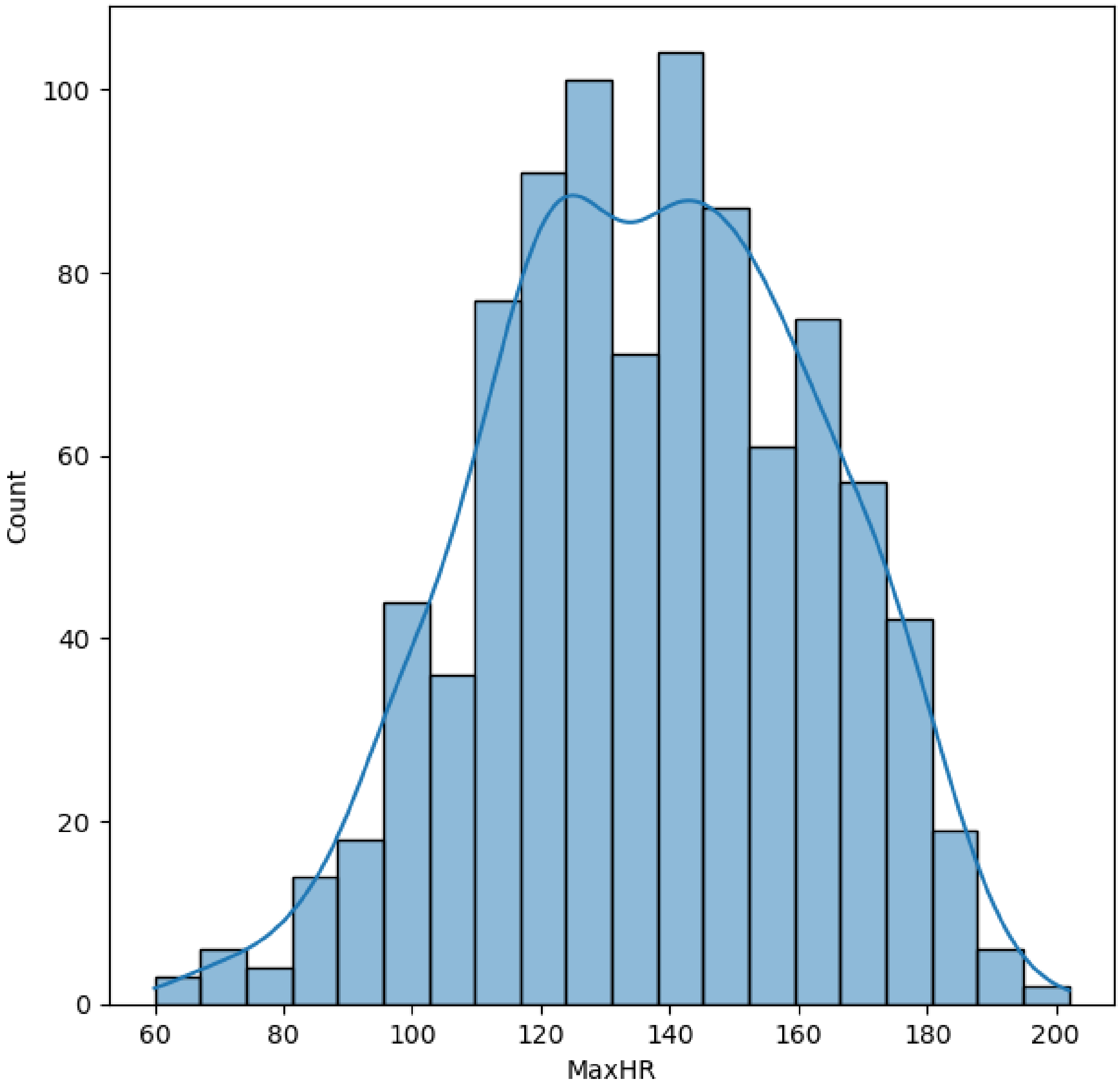


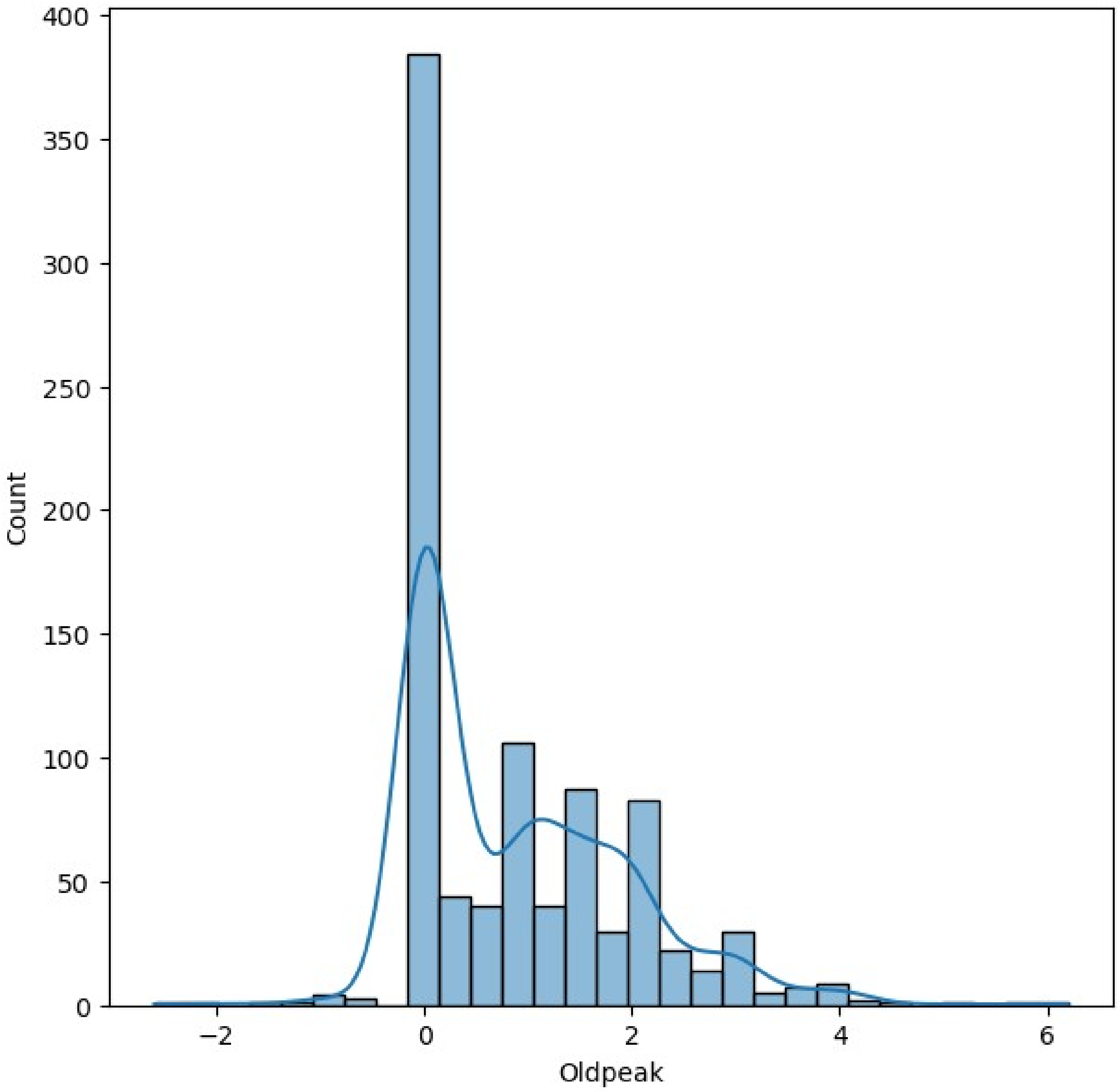
```python
for i in cols:
    plt.figure(figsize=(7,7))
    sns.histplot(x=df[i],kde=True)
```


0
20
40
60
80
100
120
Count
30
40
50
60
70
Age


Count
0
20
40
60
80
100
120
140
0
25
50
75
100
125
150
175
200
RestingBP

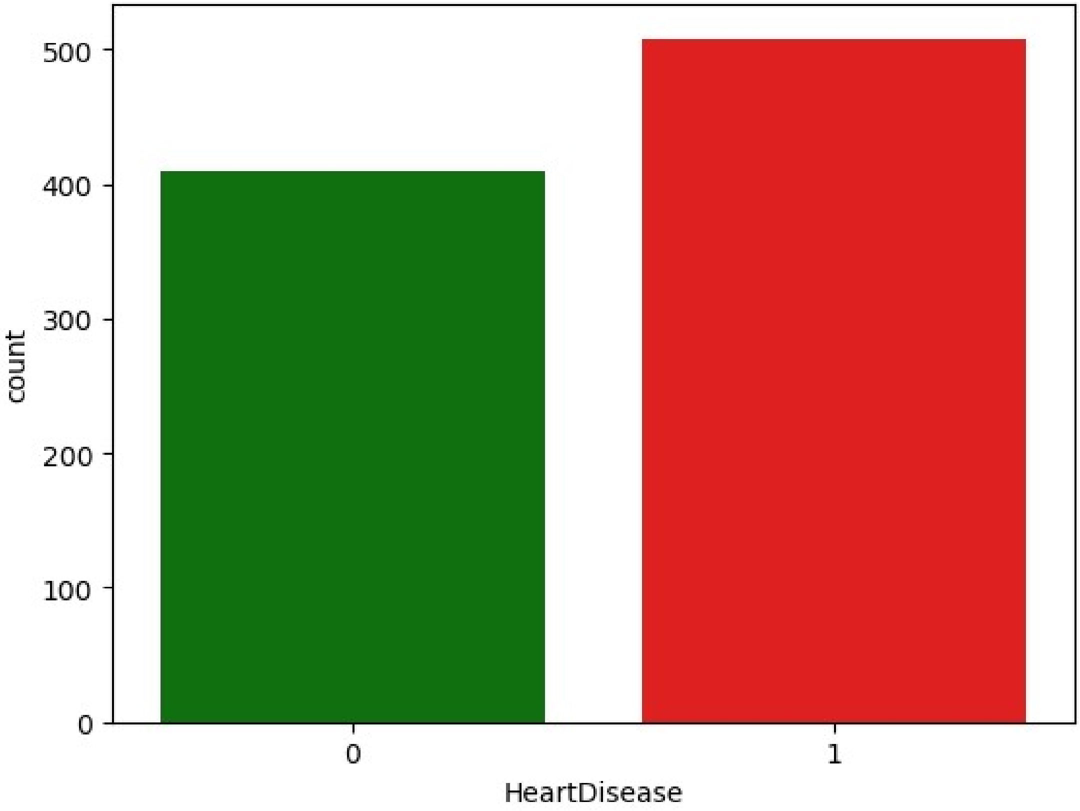

175
150
125
100
75
50
25
0
Count
0
100
200
300
400
500
600
Cholesterol


Count
0
20
40
60
80
100
60
80
100
120
140
160
180
200
MaxHR

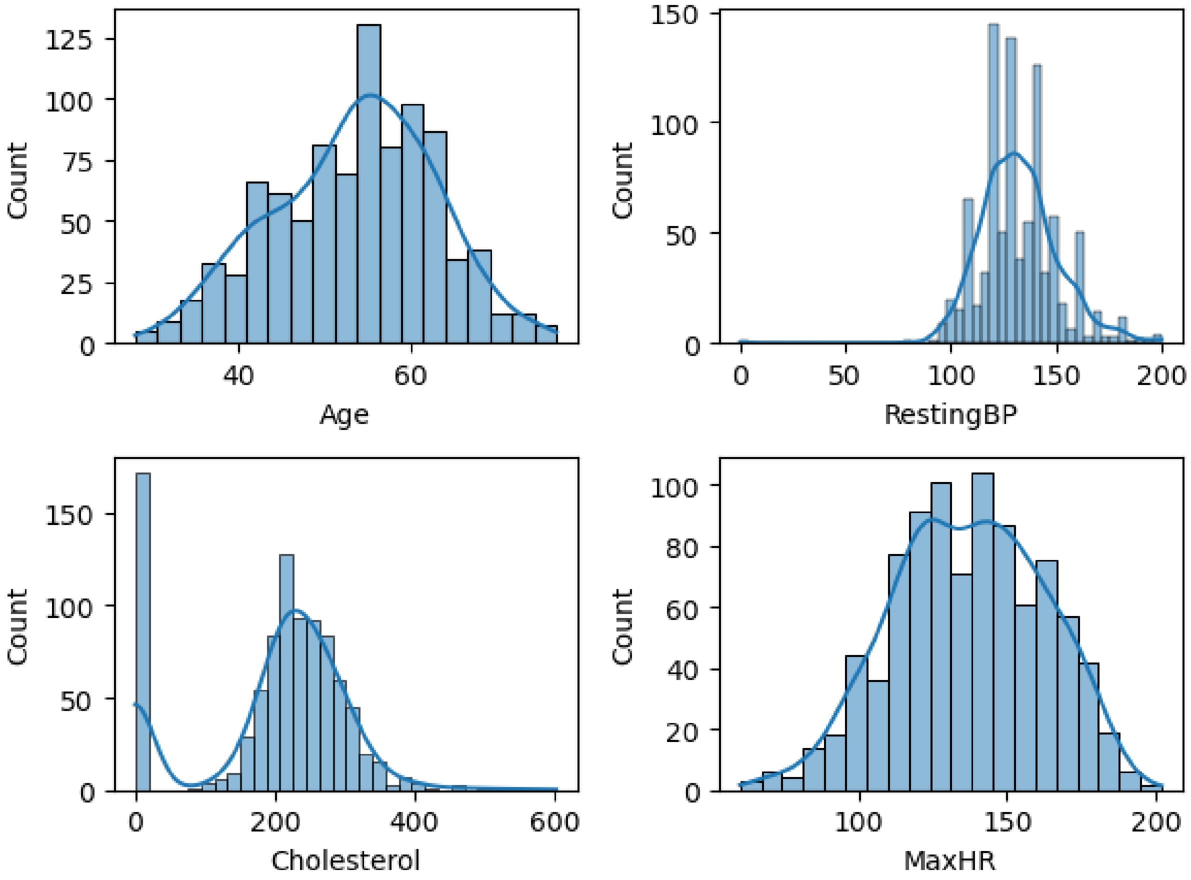


```
sns.countplot(x=df['HeartDisease'],palette=['green','red'])
```

```
<Axes: xlabel='HeartDisease', ylabel='count'>
```



```
def plotting(var,num):
    plt.subplot(2,2,num)
    sns.histplot(df[var],kde=True)

plotting('Age',1)
plotting('RestingBP',2)
plotting('Cholesterol',3)
plotting('MaxHR',4)
plt.tight_layout()
```

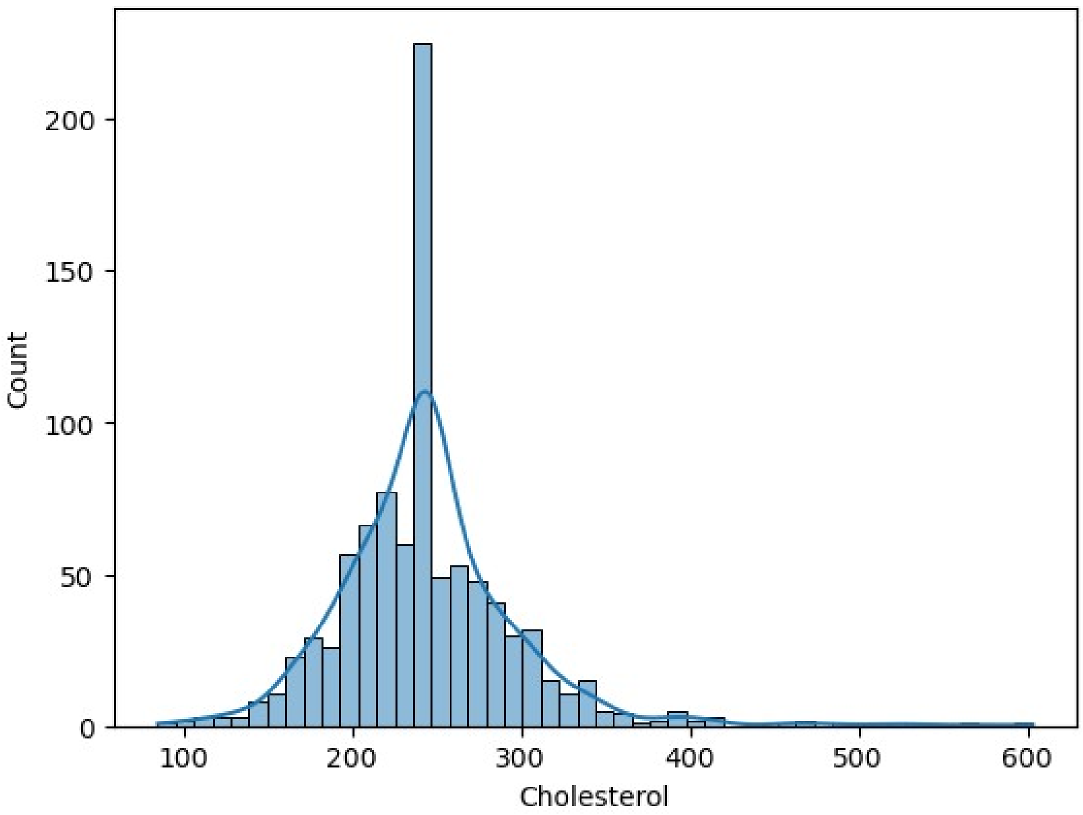


```
df_clean=df.copy()

ch_mean=df_clean.loc[df_clean['Cholesterol'] !=
0,'Cholesterol'].mean()

ch_mean
```

```
np.float64(244.6353887399464)
```

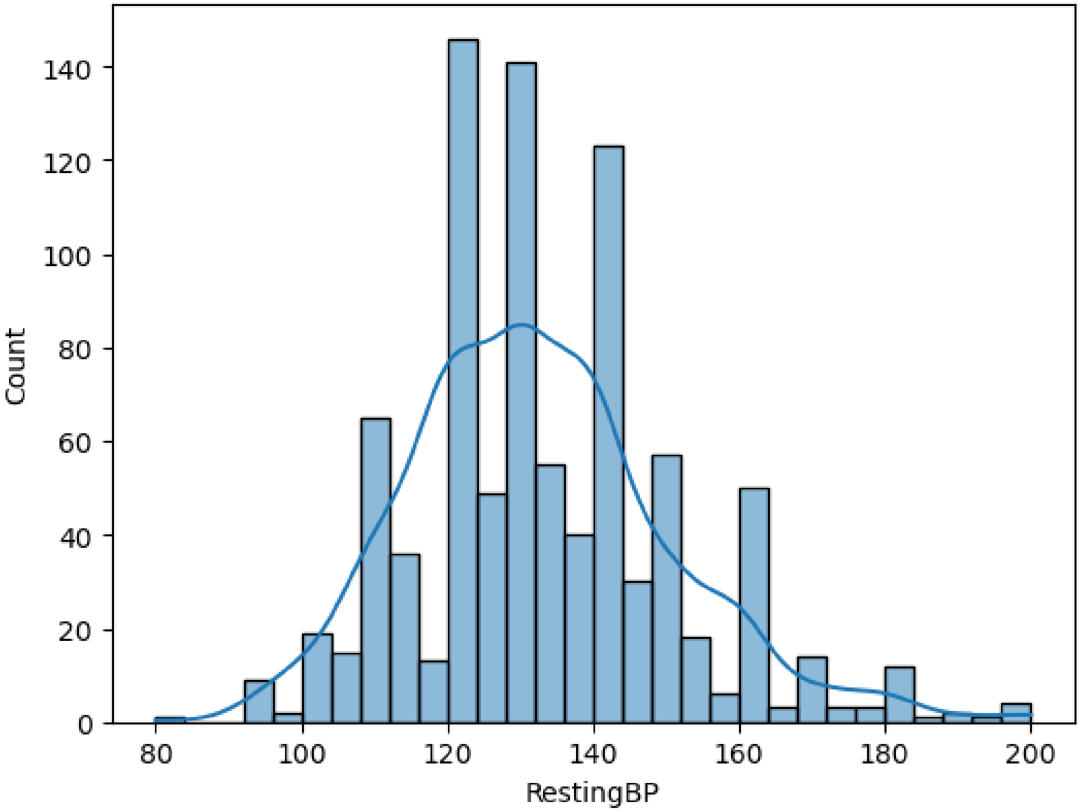
```
df_clean['Cholesterol']=df_clean['Cholesterol'].replace(0,ch_mean)
df_clean['Cholesterol']=df_clean['Cholesterol'].round(2)

ch_mean=df_clean.loc[df_clean['RestingBP'] != 0,'RestingBP'].mean()
df_clean['RestingBP']=df_clean['RestingBP'].replace(0,ch_mean)
df_clean['RestingBP']=df_clean['RestingBP'].round(2)

sns.histplot(df_clean['Cholesterol'],kde=True)
```

```
<Axes: xlabel='Cholesterol', ylabel='Count'>
```



```
sns.histplot(df_clean['RestingBP'],kde=True)
```

```
<Axes: xlabel='RestingBP', ylabel='Count'>
```



```
df_clean['Sex']=df['Sex'].map({'M':1 , 'F':0})

df_clean.rename(columns={'Sex':'is_Male'},inplace=True)

df_clean
```

```
     Age  is_Male ChestPainType  RestingBP  Cholesterol  FastingBS
RestingECG  \
0     40        1           ATA      140.0        289.0          0
Normal
1     49        0           NAP      160.0        180.0          0
Normal
2     37        1           ATA      130.0        283.0          0
ST
3     48        0           ASY      138.0        214.0          0
Normal
4     54        1           NAP      150.0        195.0          0
Normal
..   ...      ...           ...        ...          ...        ...
...
913   45        1            TA      110.0        264.0          0
Normal
914   68        1           ASY      144.0        193.0          1
Normal
915   57        1           ASY      130.0        131.0          0
```

```
Normal
916   57        0           ATA      130.0        236.0          0
LVH
917   38        1           NAP      138.0        175.0          0
Normal

     MaxHR ExerciseAngina  Oldpeak ST_Slope  HeartDisease
0      172              N      0.0       Up             0
1      156              N      1.0     Flat             1
2       98              N      0.0       Up             0
3      108              Y      1.5     Flat             1
4      122              N      0.0       Up             0
..     ...            ...      ...      ...           ...
913    132              N      1.2     Flat             1
914    141              N      3.4     Flat             1
915    115              Y      1.2     Flat             1
916    174              N      0.0     Flat             1
917    173              N      0.0       Up             0

[918 rows x 12 columns]
```

```
df_clean=pd.get_dummies(df_clean,columns=['ChestPainType'],drop_first=
False)
```

```
df_clean
```

```
     Age  is_Male  RestingBP  Cholesterol  FastingBS RestingECG  MaxHR
\
0     40        1      140.0        289.0          0     Normal    172

1     49        0      160.0        180.0          0     Normal    156

2     37        1      130.0        283.0          0         ST     98

3     48        0      138.0        214.0          0     Normal    108

4     54        1      150.0        195.0          0     Normal    122

..   ...      ...        ...          ...        ...        ...    ...

913   45        1      110.0        264.0          0     Normal    132

914   68        1      144.0        193.0          1     Normal    141

915   57        1      130.0        131.0          0     Normal    115

916   57        0      130.0        236.0          0        LVH    174

917   38        1      138.0        175.0          0     Normal    173
```

```
    ExerciseAngina  Oldpeak ST_Slope  HeartDisease  ChestPainType_ASY  
\
0                N      0.0       Up             0              False   

1                N      1.0     Flat             1              False   

2                N      0.0       Up             0              False   

3                Y      1.5     Flat             1               True   

4                N      0.0       Up             0              False   

..             ...      ...      ...           ...                ...   

913              N      1.2     Flat             1              False   

914              N      3.4     Flat             1               True   

915              Y      1.2     Flat             1               True   

916              N      0.0     Flat             1              False   

917              N      0.0       Up             0              False   


     ChestPainType_ATA  ChestPainType_NAP  ChestPainType_TA  
0                 True              False             False  
1                False               True             False  
2                 True              False             False  
3                False              False             False  
4                False               True             False  
..                 ...                ...               ...  
913              False              False              True  
914              False              False             False  
915              False              False             False  
916               True              False             False  
917              False               True             False  

[918 rows x 15 columns]
```

```
df_clean=pd.get_dummies(df_clean,columns=['RestingECG'],drop_first=False)

df_clean=pd.get_dummies(df_clean,columns=['ExerciseAngina'],drop_first=False)

df_clean=pd.get_dummies(df_clean,columns=['ST_Slope'],drop_first=False)

df_clean
```

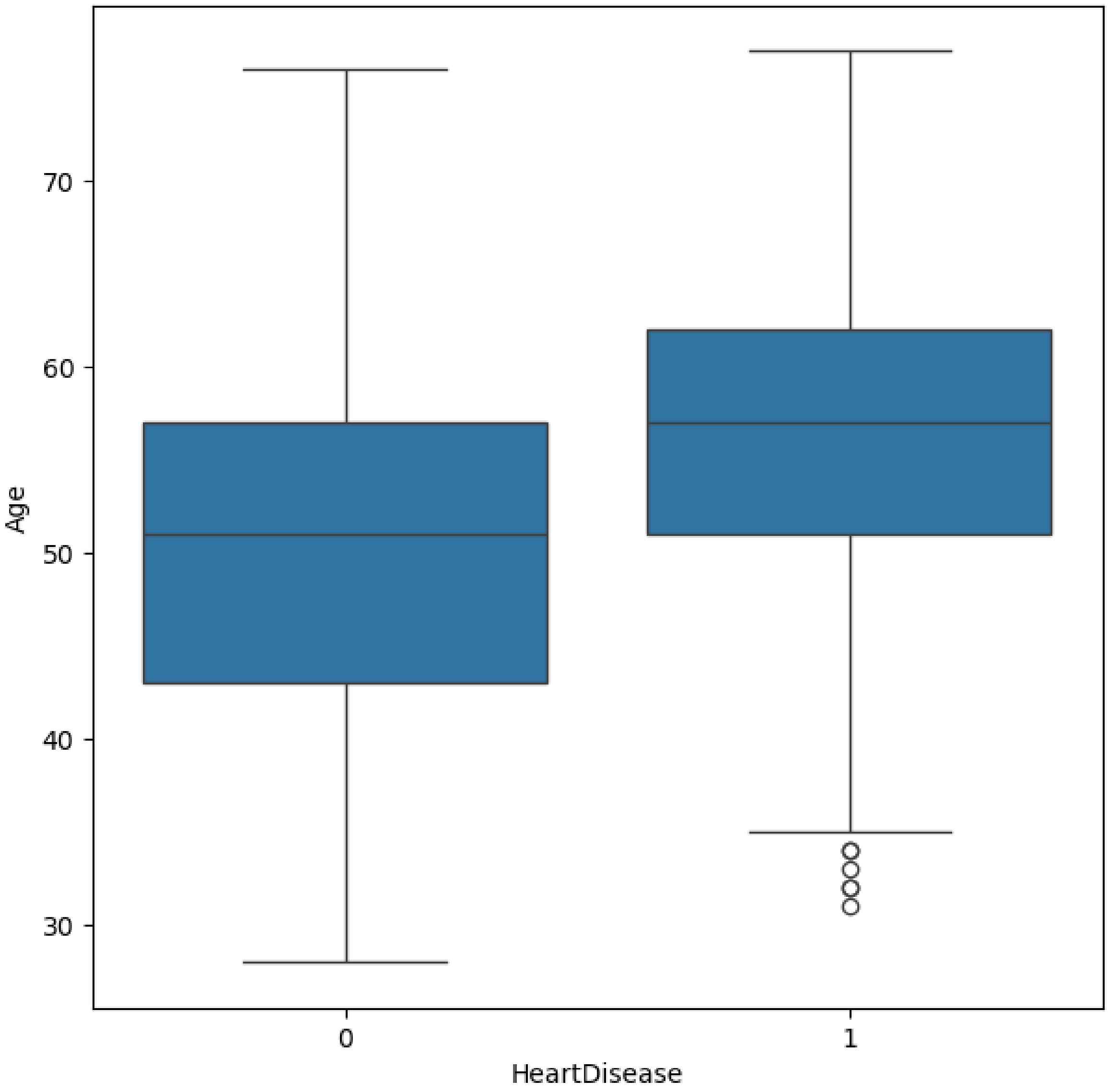
```
     Age  is_Male  RestingBP  Cholesterol  FastingBS  MaxHR
Oldpeak  \
0      40        1      140.0        289.0          0    172       0.0

1      49        0      160.0        180.0          0    156       1.0

2      37        1      130.0        283.0          0     98       0.0

3      48        0      138.0        214.0          0    108       1.5

4      54        1      150.0        195.0          0    122       0.0

..   ...      ...        ...          ...        ...    ...       ...

913    45        1      110.0        264.0          0    132       1.2

914    68        1      144.0        193.0          1    141       3.4

915    57        1      130.0        131.0          0    115       1.2

916    57        0      130.0        236.0          0    174       0.0

917    38        1      138.0        175.0          0    173       0.0


     HeartDisease  ChestPainType_ASY  ChestPainType_ATA
ChestPainType_NAP  \
0               0              False               True
False
1               1              False              False
True
2               0              False               True
False
3               1               True              False
False
4               0              False              False
True
..            ...                ...                ...
...
913             1              False              False
False
914             1               True              False
False
915             1               True              False
False
916             1              False               True
False
917             0              False              False
True

     ChestPainType_TA  RestingECG_LVH  RestingECG_Normal
```

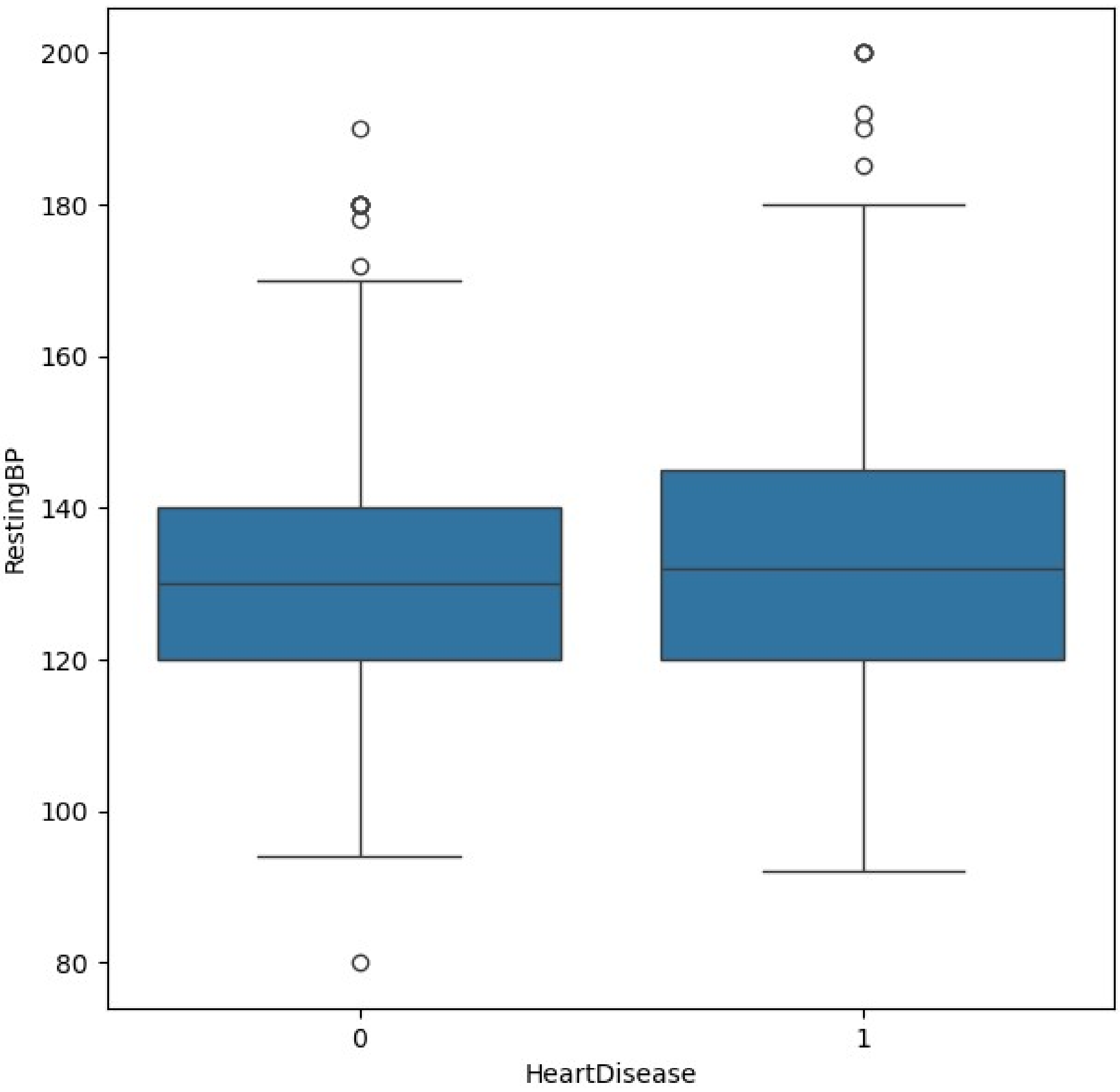
```
RestingECG_ST  \
0               False           False               True
False
1               False           False               True
False
2               False           False              False
True
3               False           False               True
False
4               False           False               True
False
..                ...             ...                ...            ..
.
913              True           False               True
False
914             False           False               True
False
915             False           False               True
False
916             False            True              False
False
917             False           False               True
False

     ExerciseAngina_N  ExerciseAngina_Y  ST_Slope_Down  ST_Slope_Flat
\
0                True             False          False          False

1                True             False          False           True

2                True             False          False          False

3               False              True          False           True

4                True             False          False          False

..                ...               ...            ...            ...

913              True             False          False           True

914              True             False          False           True

915             False              True          False           True

916              True             False          False           True

917              True             False          False          False


     ST_Slope_Up
0           True
```

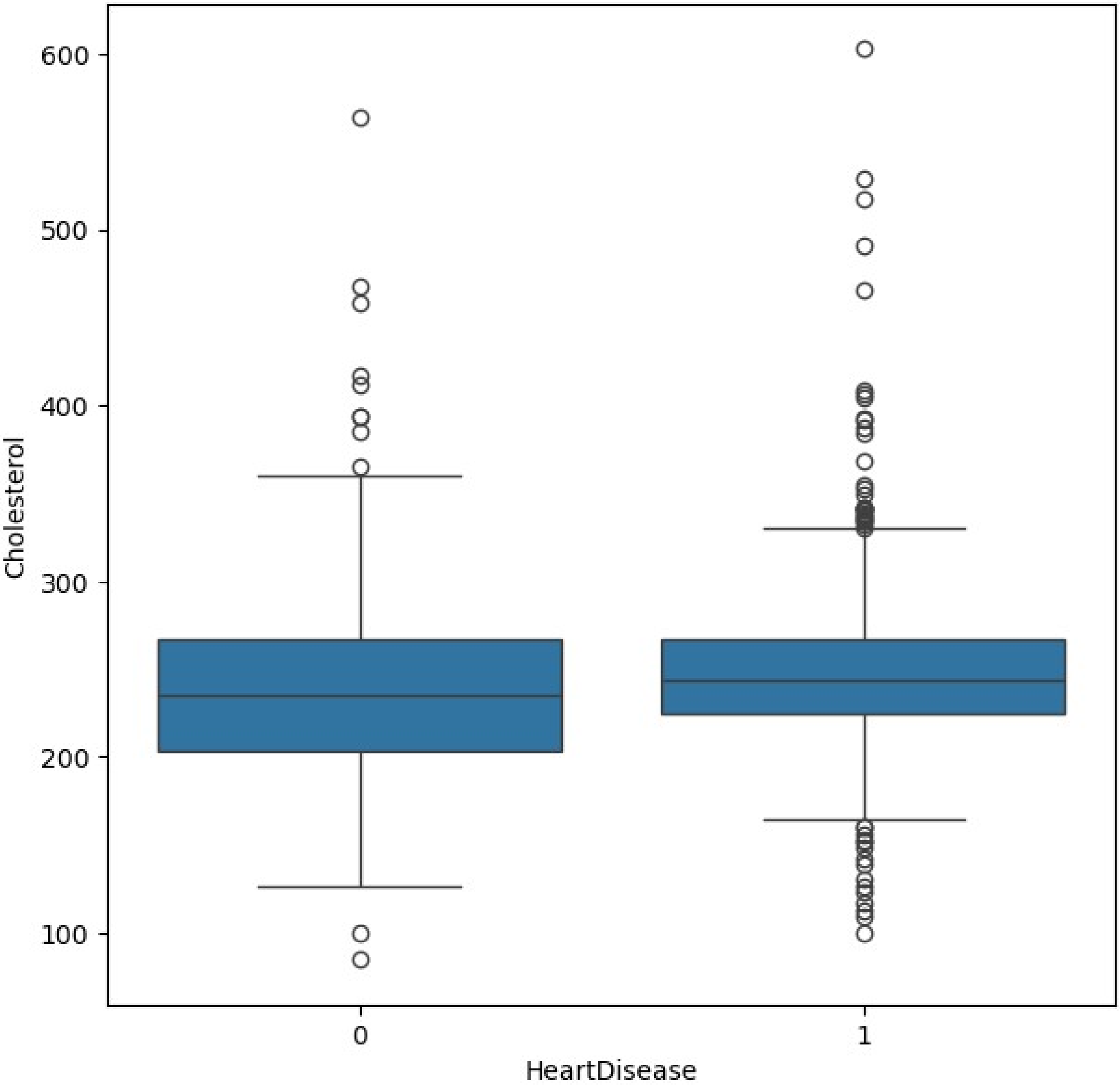
```
1           False
2            True
3           False
4            True
..            ...
913         False
914         False
915         False
916         False
917          True

[918 rows x 20 columns]
```

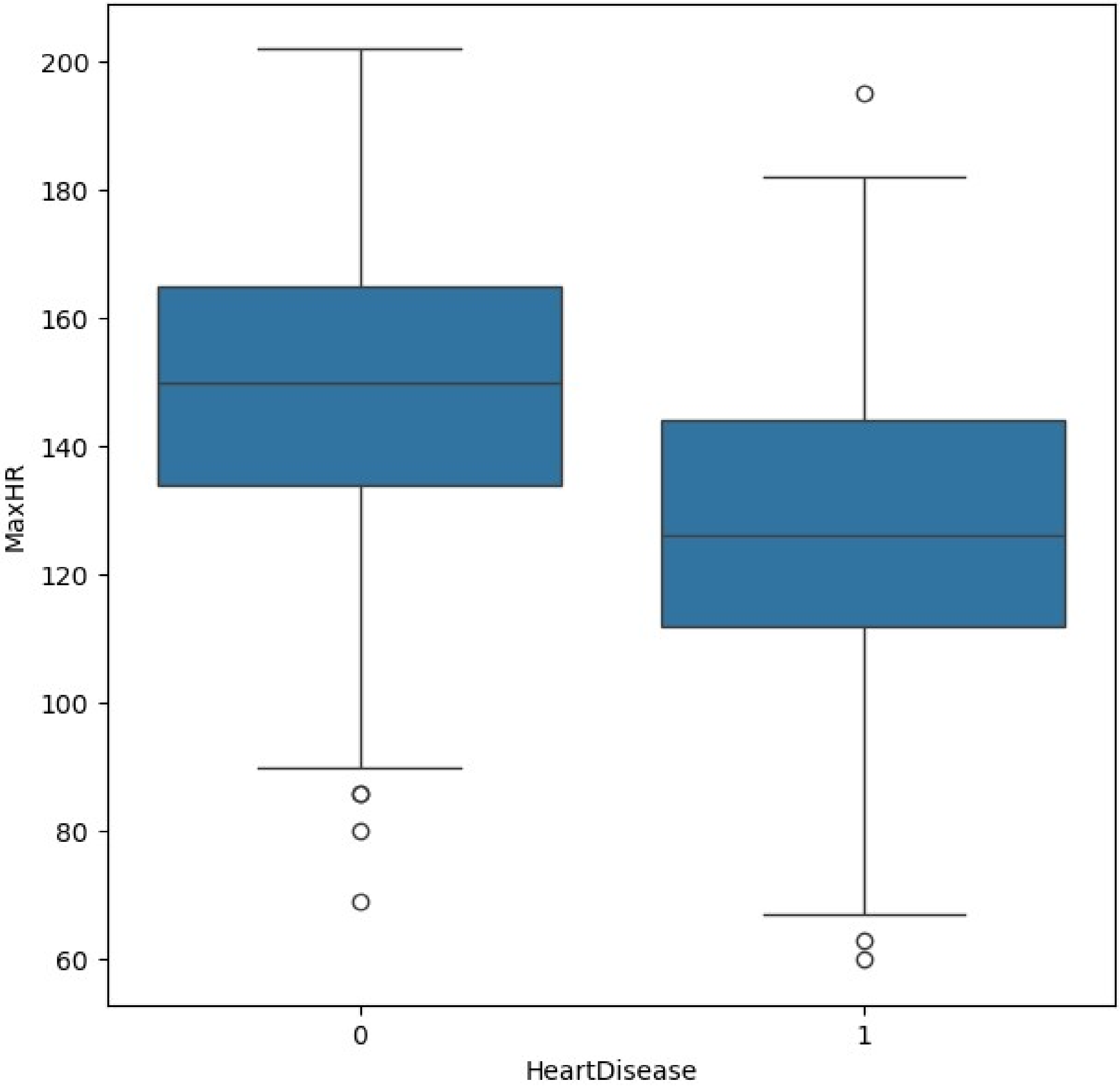
```
df_clean=df_clean.astype(int)
```

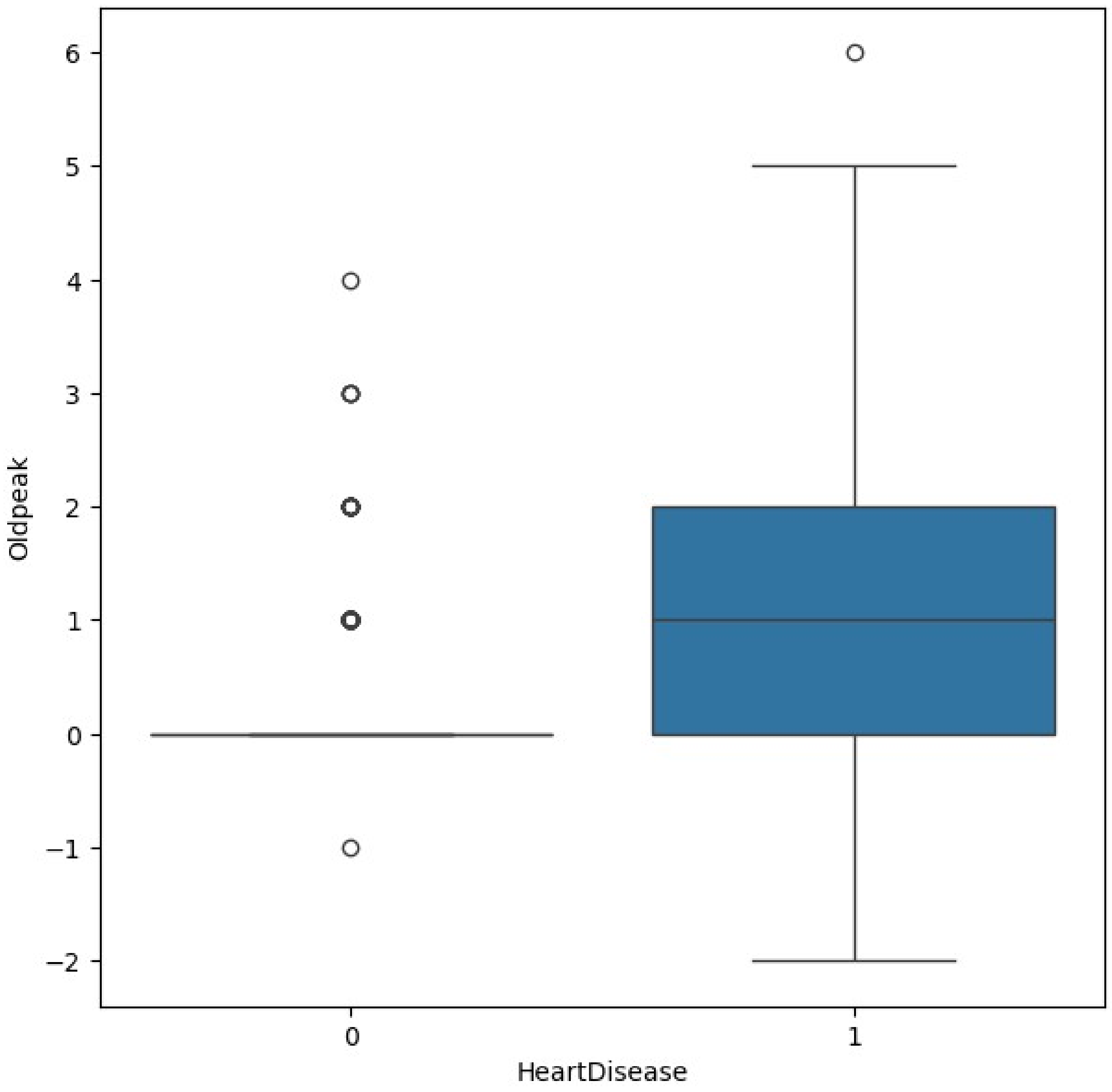
```
 numeric_cols=['Age', 'RestingBP', 'Cholesterol', 'MaxHR', 'Oldpeak', 
'HeartDisease']
for i in numeric_cols:
    plt.figure(figsize=(7,7))
    sns.boxplot(y=i,x='HeartDisease',data=df_clean)
```

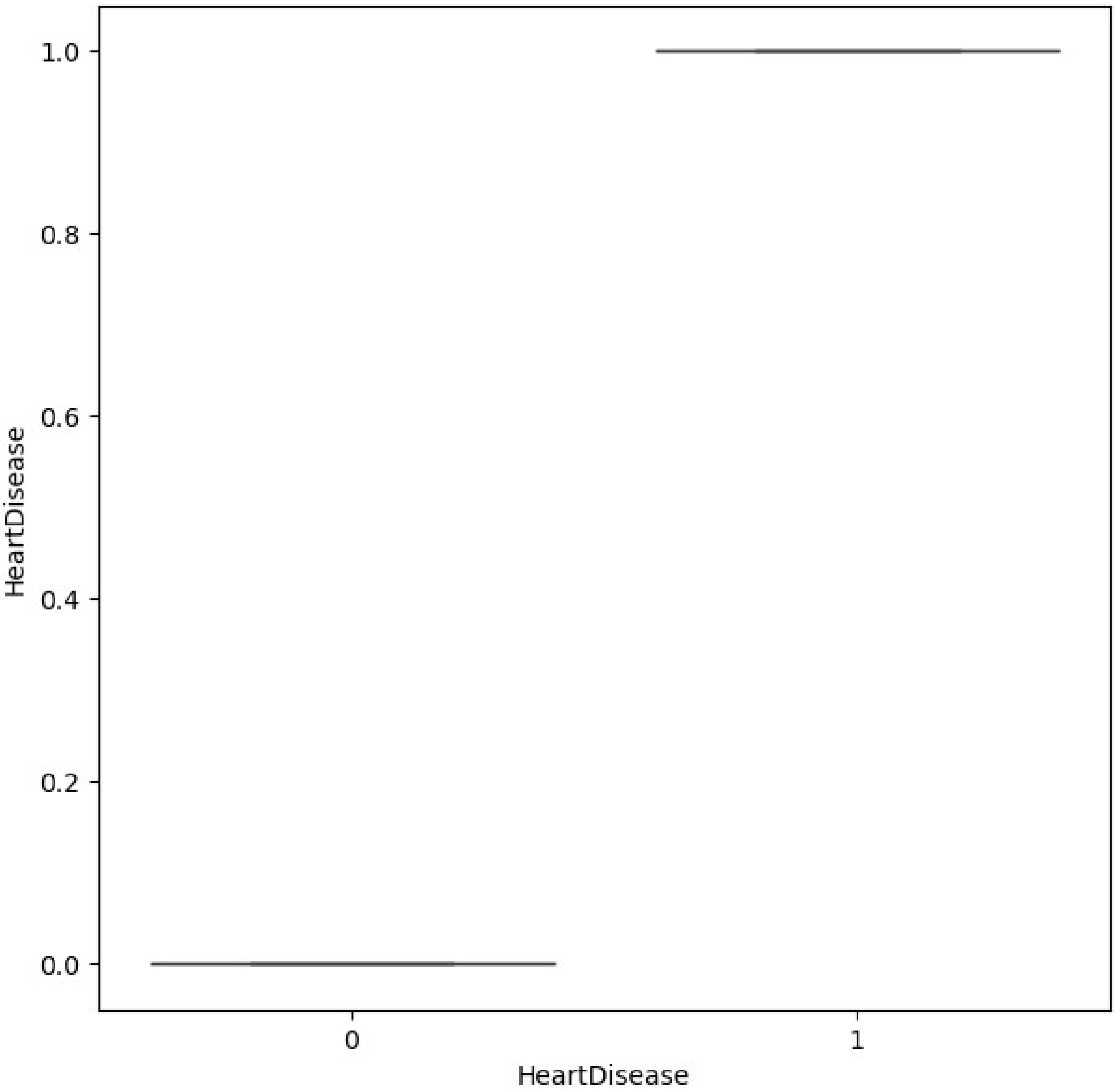

70
60
50
40
30
Age
0
1
HeartDisease


200
180
160
140
120
100
80
RestingBP
0
1
HeartDisease

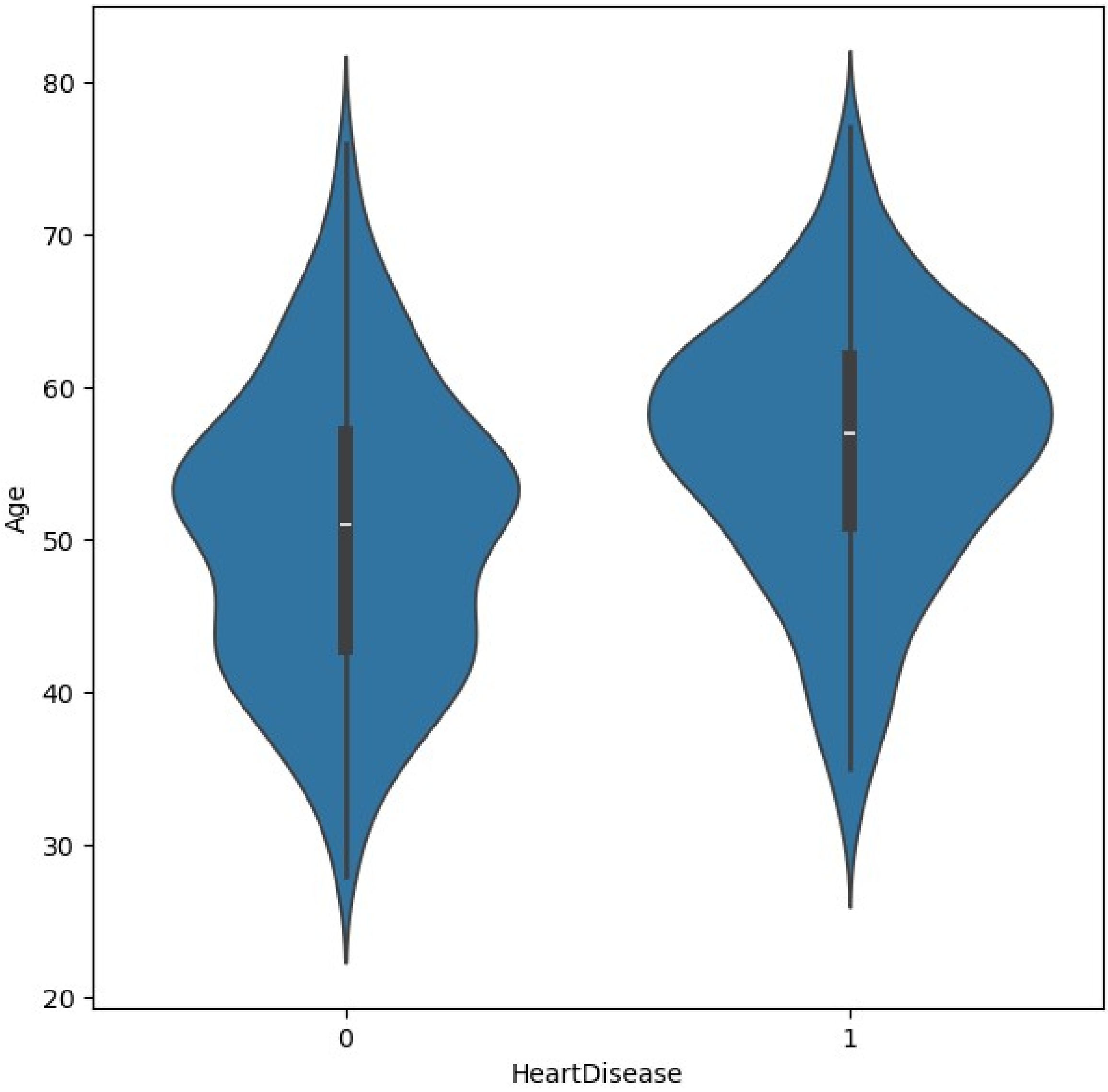

600
500
400
300
200
100
Cholesterol
0
1
HeartDisease

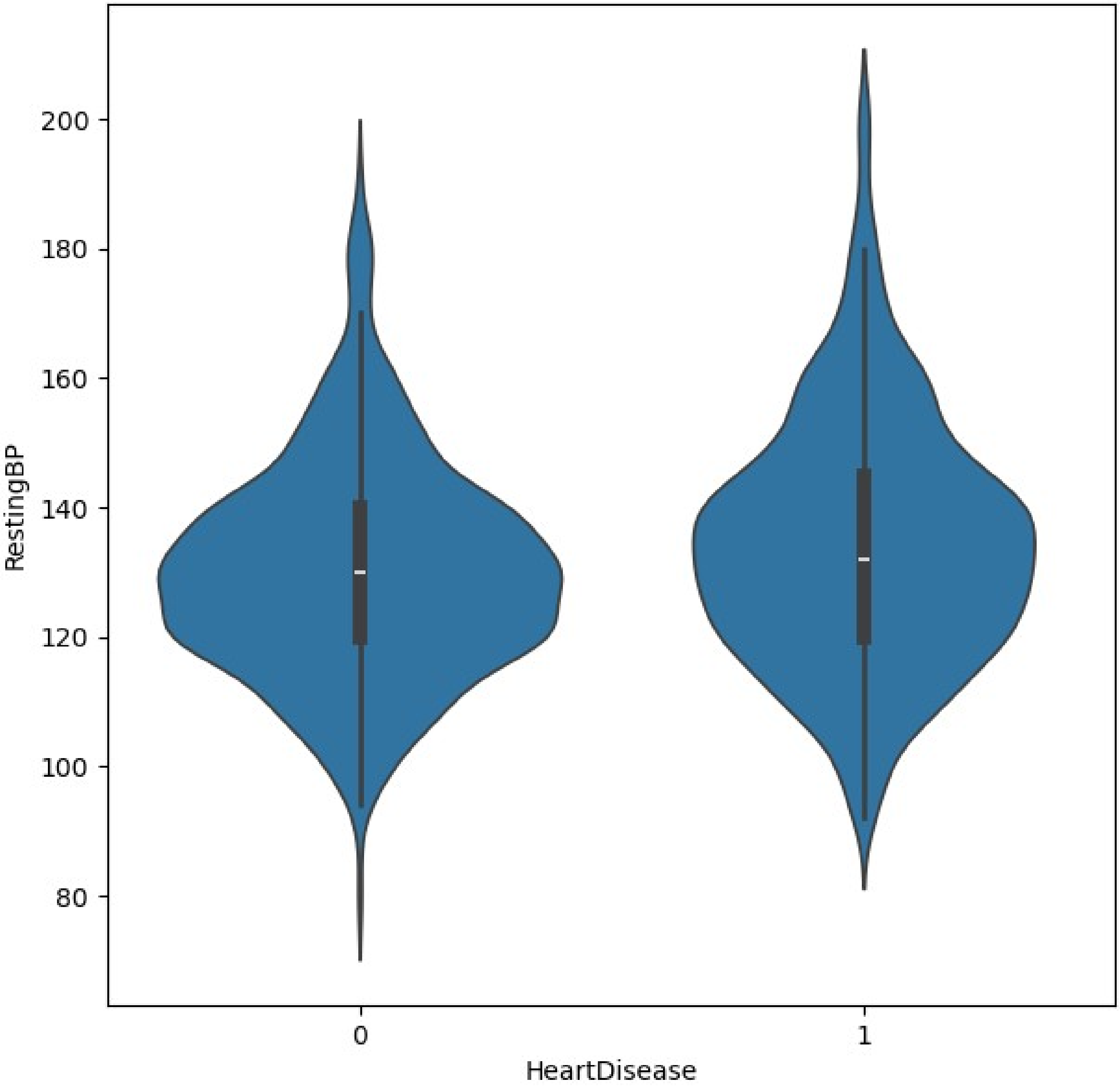

200
180
160
140
120
100
80
60
MaxHR
0
1
HeartDisease

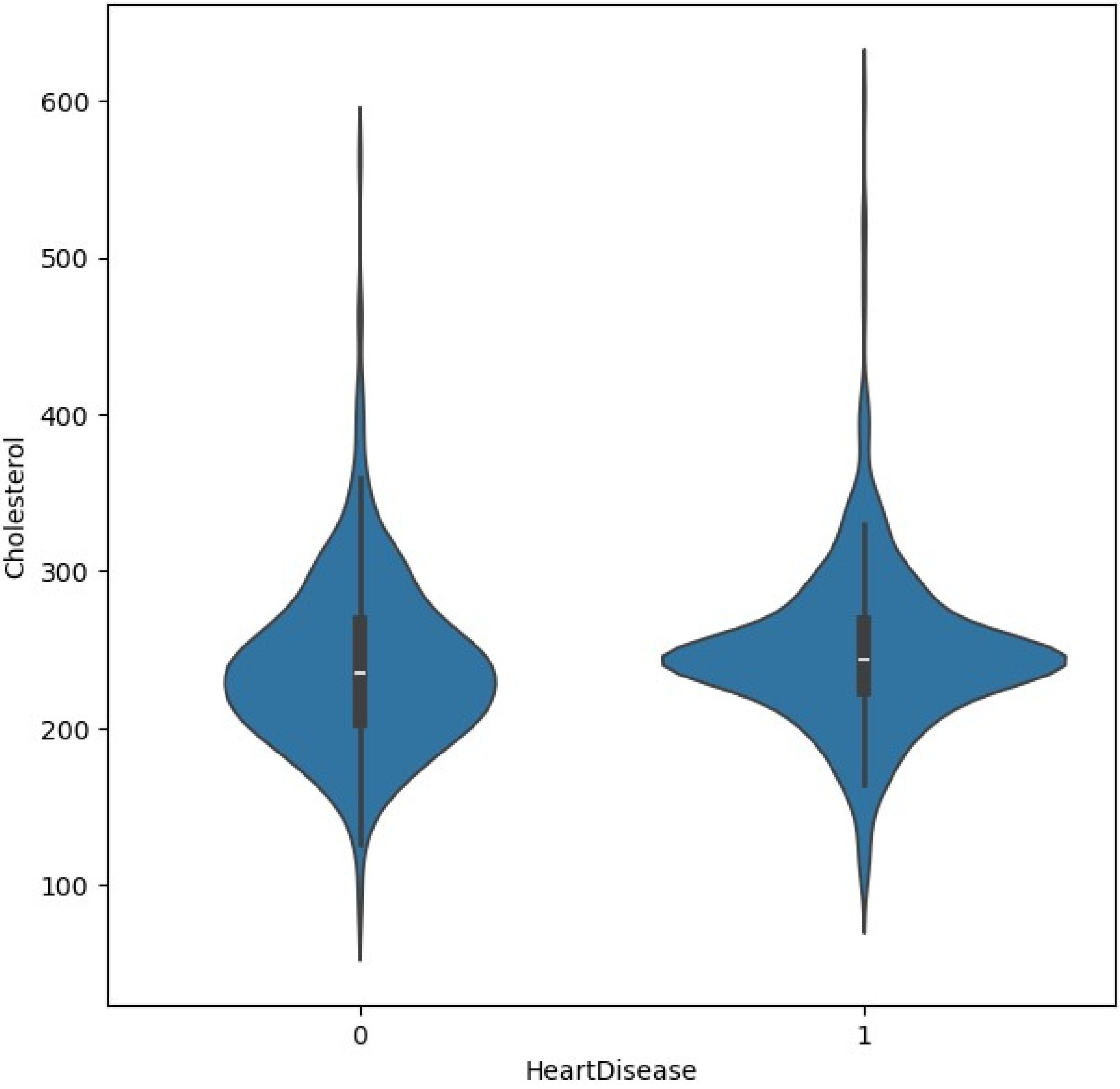

6
5
4
3
2
1
0
−1
−2
Oldpeak
0
1
HeartDisease

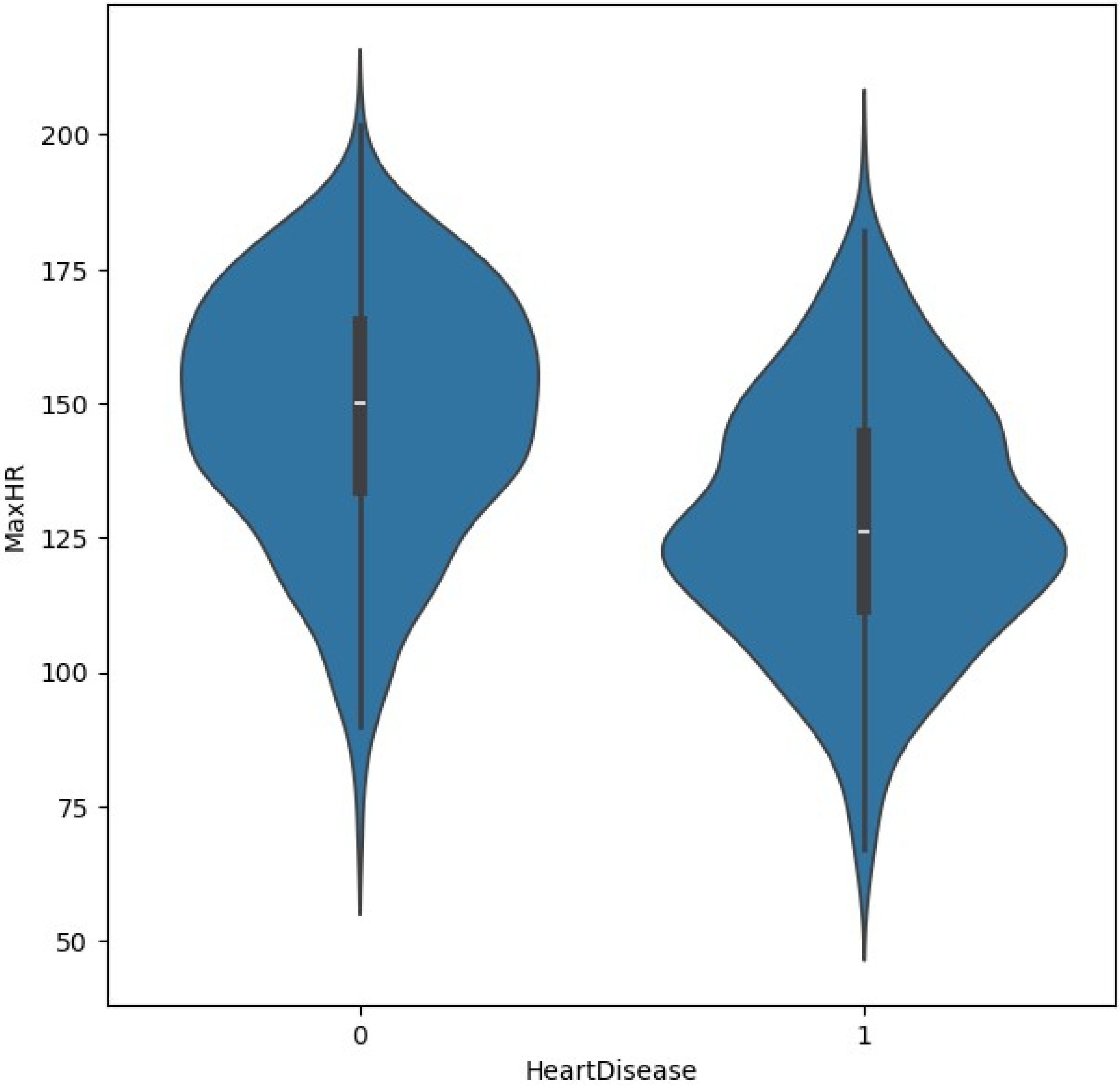


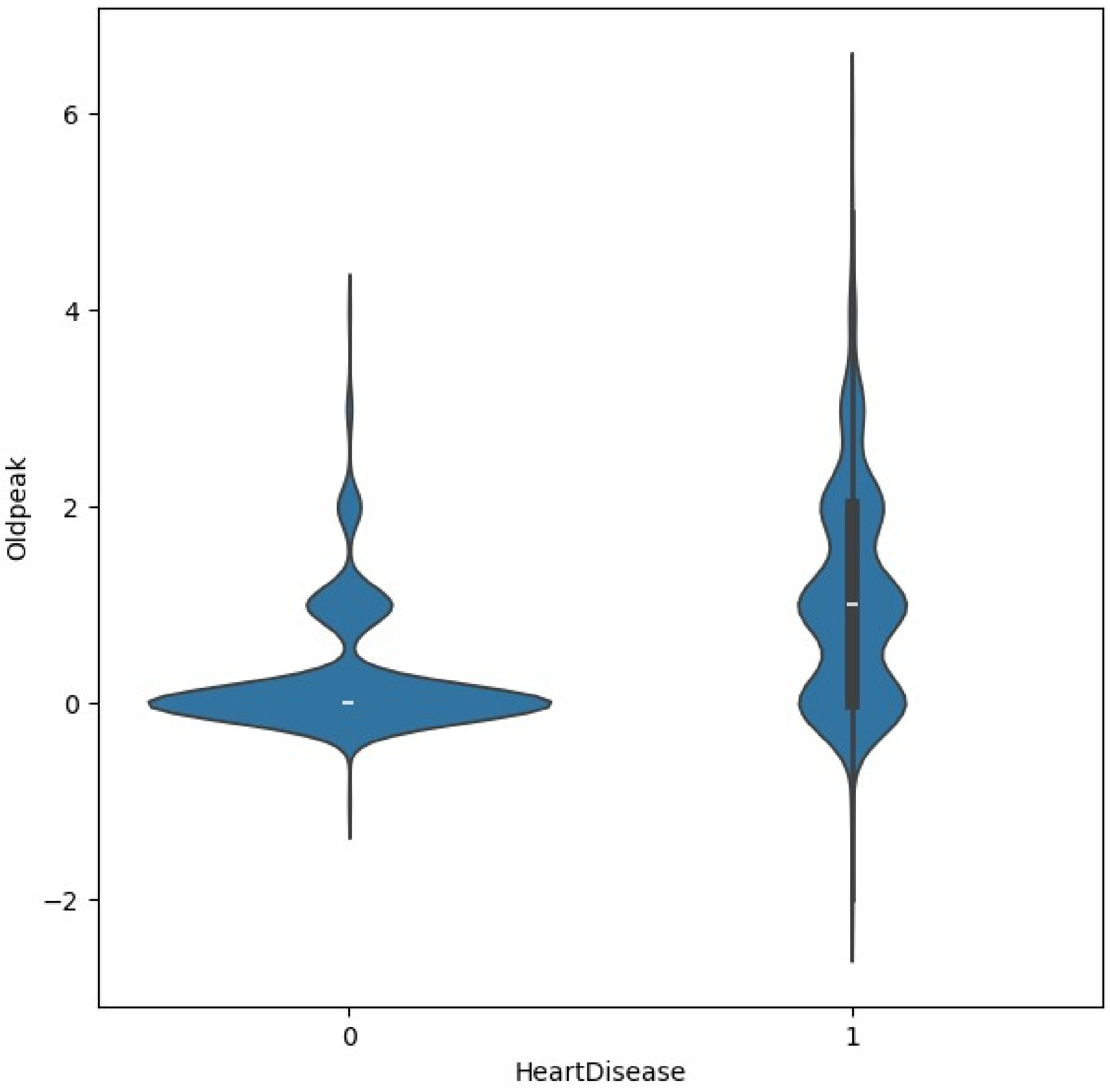
```
 numeric_cols=['Age', 'RestingBP', 'Cholesterol', 'MaxHR', 'Oldpeak',
'HeartDisease']
for i in numeric_cols:
    plt.figure(figsize=(7,7))
    sns.violinplot(y=i,x='HeartDisease',data=df_clean)
```

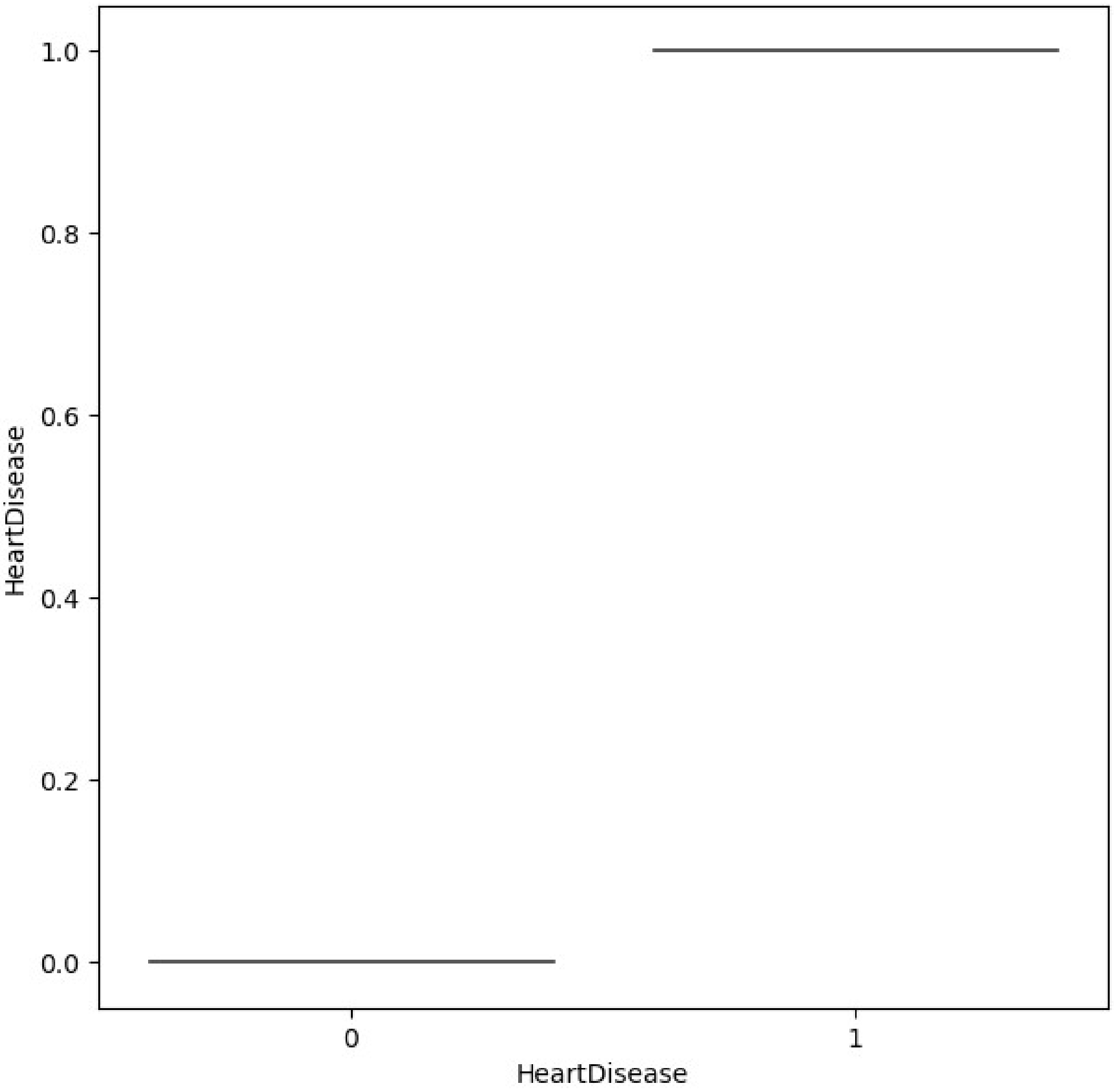

Age
80
70
60
50
40
30
20
0
1
HeartDisease


RestingBP
200
180
160
140
120
100
80
0
1
HeartDisease


600
500
400
300
200
100
Cholesterol
0
1
HeartDisease

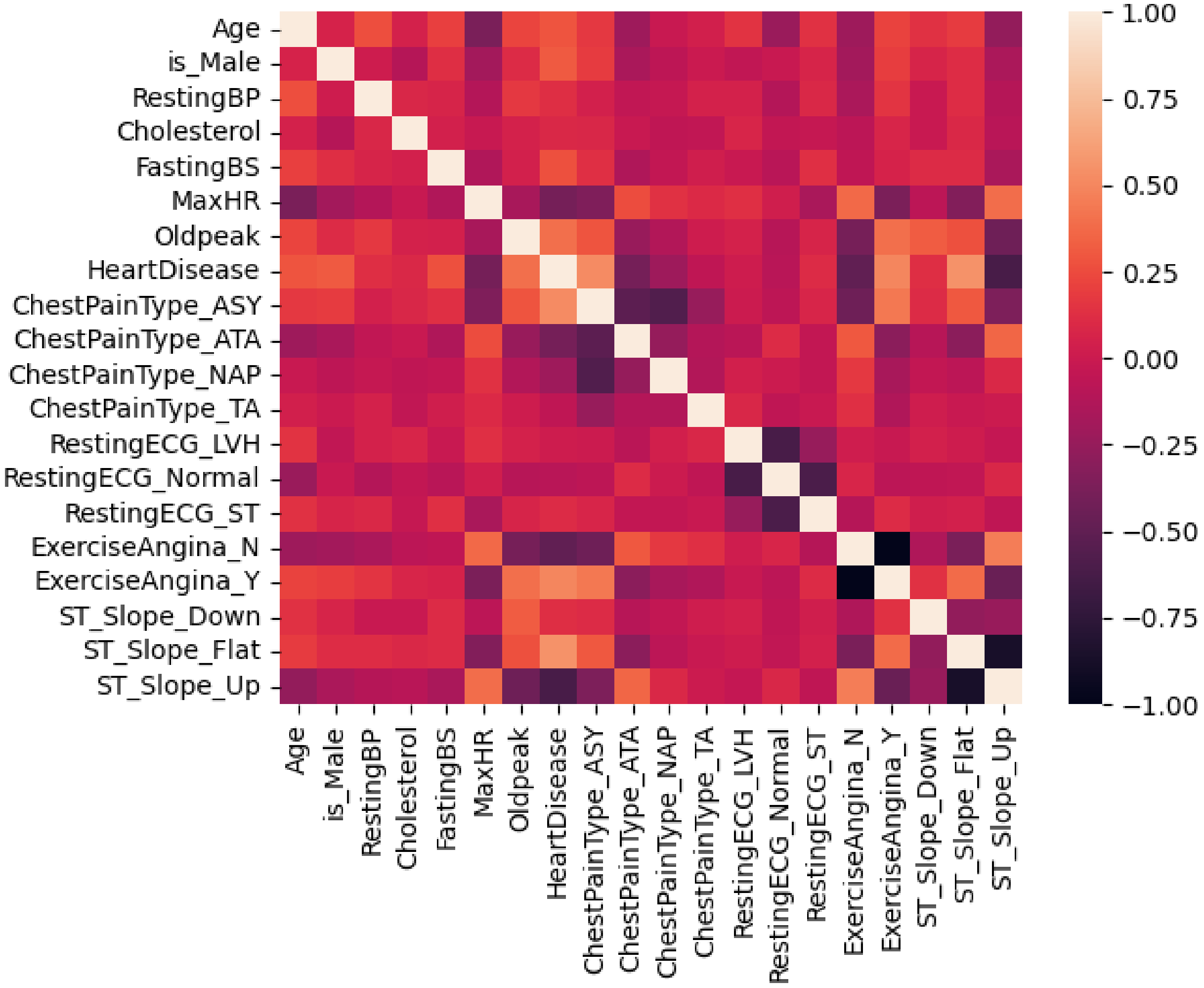

200
175
150
125
100
75
50
MaxHR
0
1
HeartDisease


6
4
2
0
−2
Oldpeak
0
1
HeartDisease



```
# from sklearn.preprocessing import StandardScaler
# cols=['Age','RestingBP','Cholesterol','MaxHR']
# scaler=StandardScaler()
# df_clean[cols]=scaler.fit_transform(df_clean[cols])
# df_clean

sns.heatmap(df_clean.corr())
```

```
<Axes: >
```



```
df_corr=df_clean.corr()['HeartDisease'].sort_values(ascending=False)

df_corr[df_corr>0]

HeartDisease         1.000000
ST_Slope_Flat        0.554134
ChestPainType_ASY    0.516716
ExerciseAngina_Y     0.494282
Oldpeak              0.392385
is_Male              0.305445
Age                  0.282039
FastingBS            0.267291
ST_Slope_Down        0.122527
RestingBP            0.117909
RestingECG_ST        0.102527
Cholesterol          0.092586
RestingECG_LVH       0.010670
Name: HeartDisease, dtype: float64
```

```
df_clean.columns
```

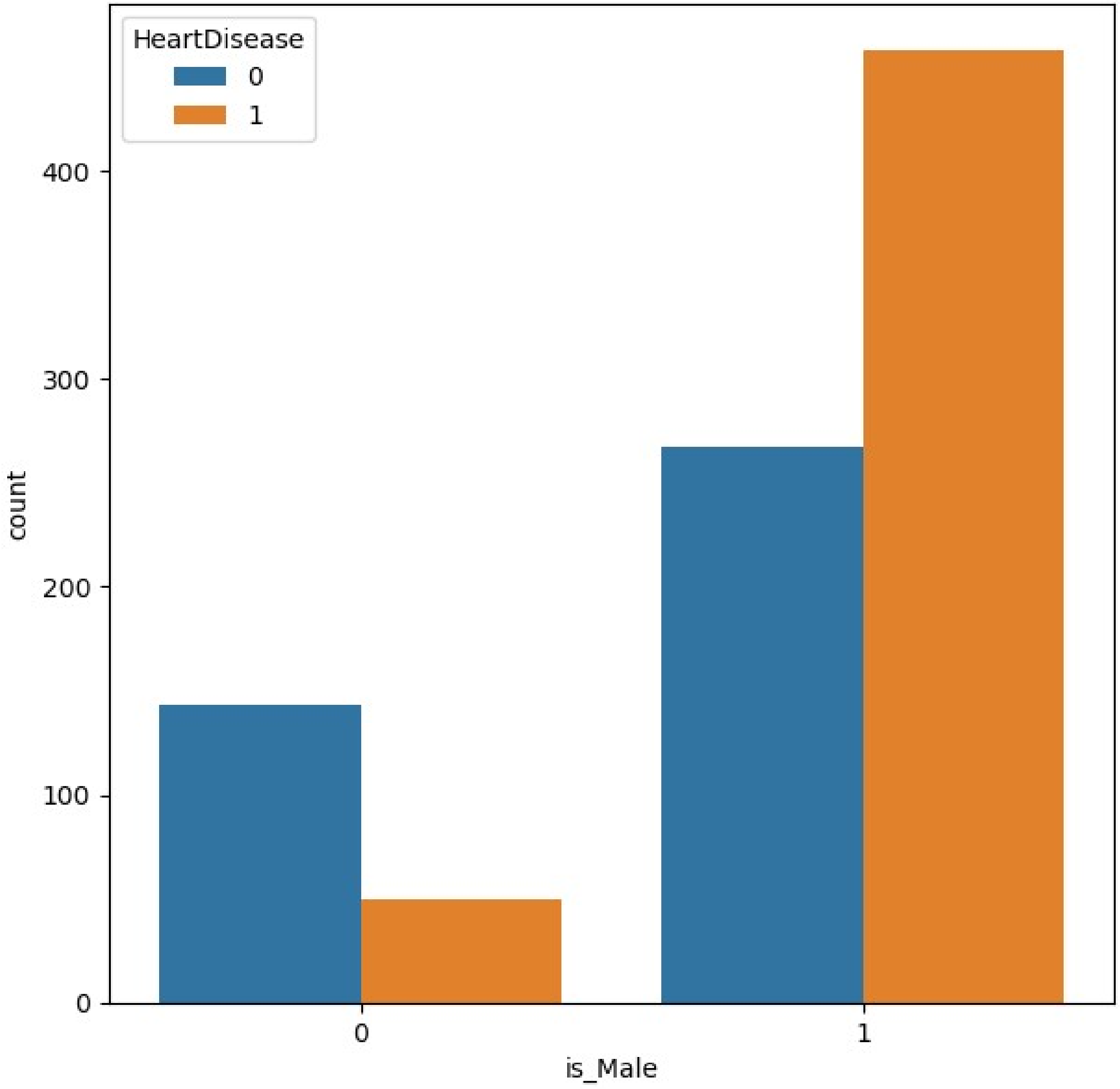
```
Index(['Age', 'is_Male', 'RestingBP', 'Cholesterol', 'FastingBS',
'MaxHR',
       'Oldpeak', 'HeartDisease', 'ChestPainType_ASY',
'ChestPainType_ATA',
       'ChestPainType_NAP', 'ChestPainType_TA', 'RestingECG_LVH',
       'RestingECG_Normal', 'RestingECG_ST', 'ExerciseAngina_N',
       'ExerciseAngina_Y', 'ST_Slope_Down', 'ST_Slope_Flat',
'ST_Slope_Up'],
      dtype='object')
```

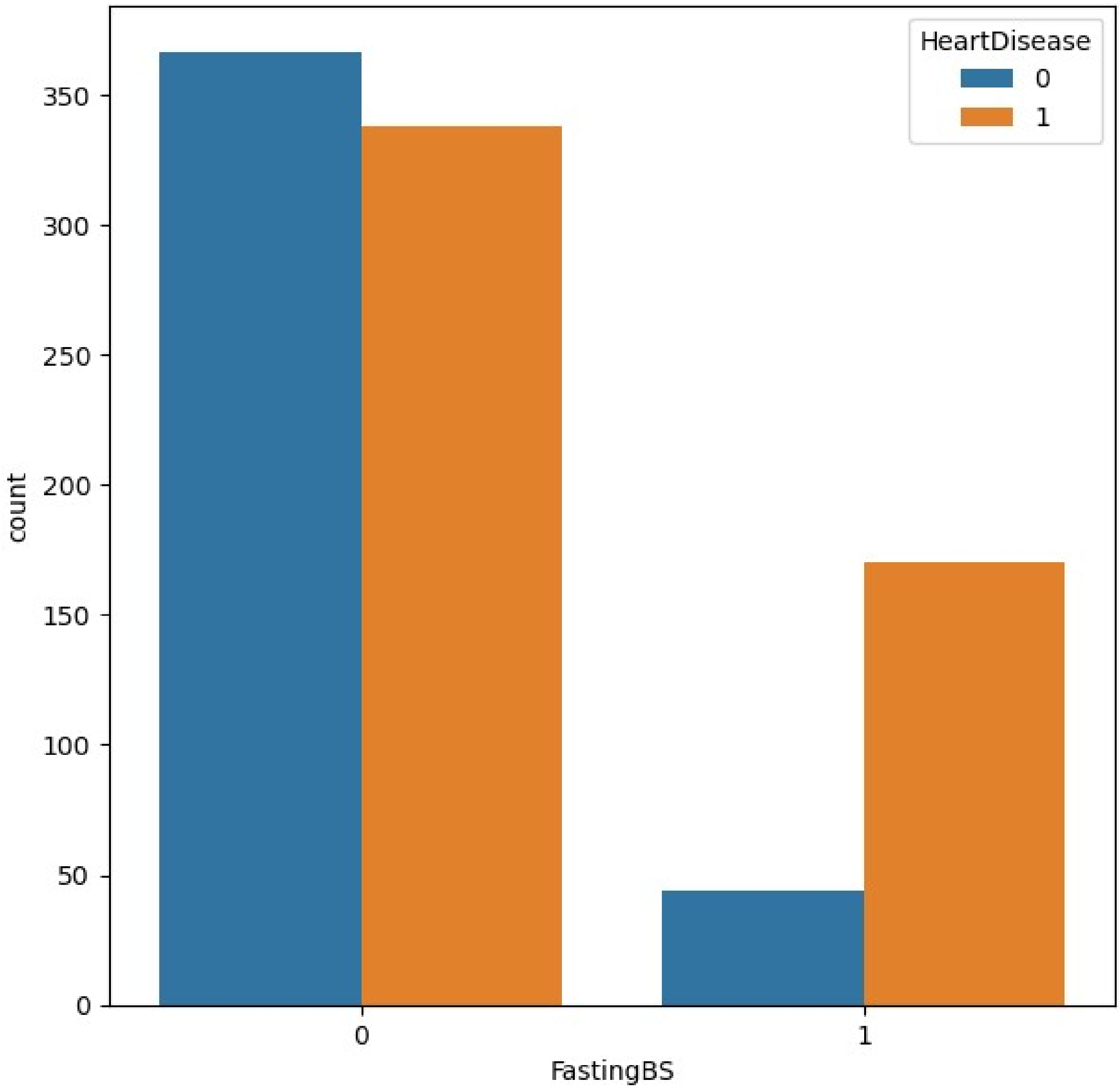
```
cat_features=[ 'is_Male', 'FastingBS','ChestPainType_ASY',
         'ChestPainType_ATA', 'ChestPainType_NAP',
       'ChestPainType_TA', 'RestingECG_LVH', 'RestingECG_Normal',
'RestingECG_ST','ExerciseAngina_N',
       'ExerciseAngina_Y',  'ST_Slope_Down','ST_Slope_Flat',
'ST_Slope_Up']

from scipy.stats import chi2_contingency

alpha = 0.05

df_clean['HeartDisease_bin'] = df_clean['HeartDisease']
chi2_results = {}

for col in cat_features:
    contingency = pd.crosstab(df_clean[col],
df_clean['HeartDisease_bin'])
    chi2_stat, p_val, _, _ = chi2_contingency(contingency)
    decision = 'Reject Null (Keep Feature)' if p_val < alpha else
'Accept Null (Drop Feature)'
    chi2_results[col] = {
        'chi2_statistic': chi2_stat,
        'p_value': p_val,
        'Decision': decision
    }

chi2_df = pd.DataFrame(chi2_results).T
chi2_df = chi2_df.sort_values(by='p_value')
chi2_df
```

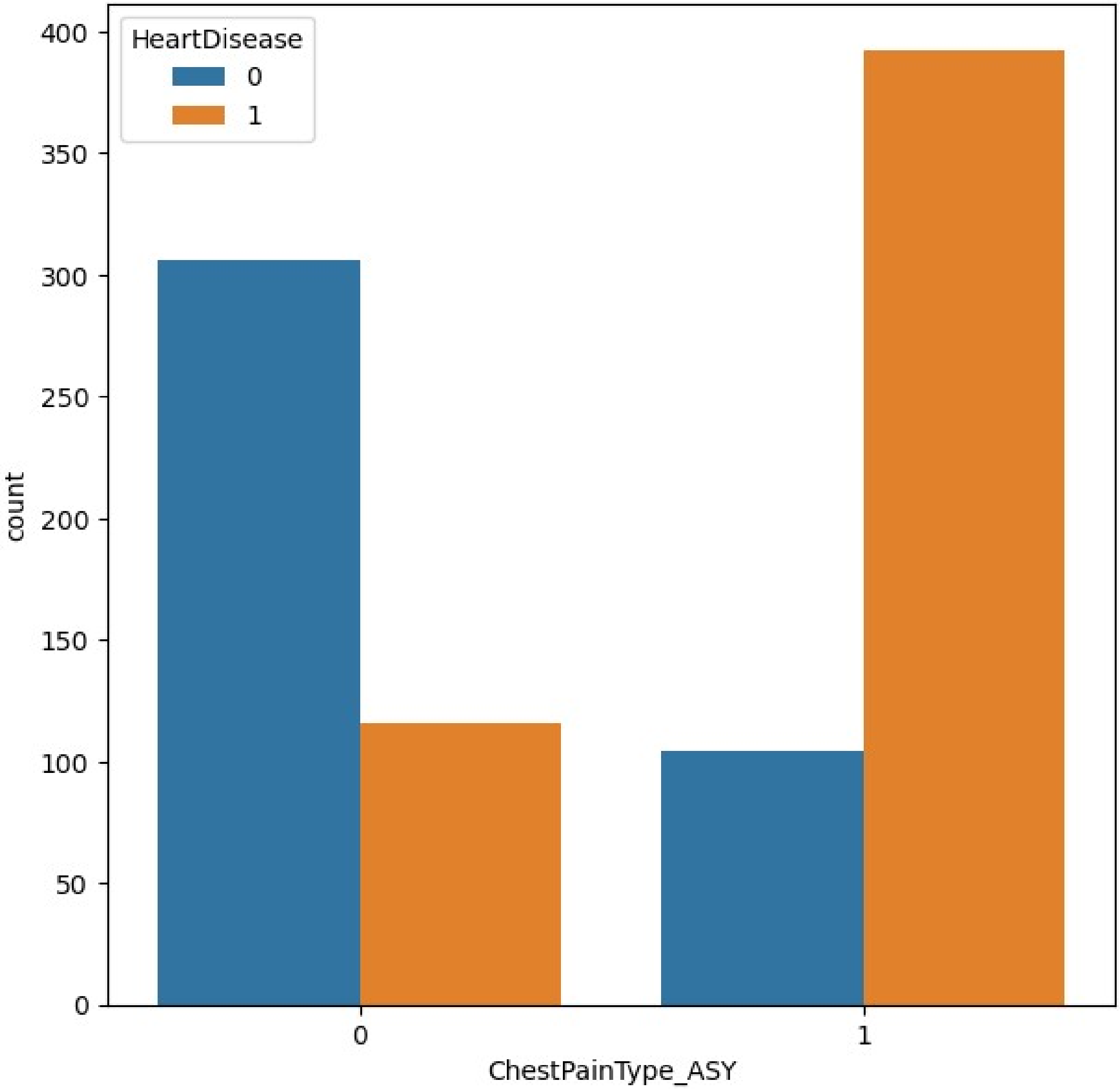
```
                  chi2_statistic   p_value                    Decision
ST_Slope_Up           352.823905       0.0  Reject Null (Keep Feature)
ST_Slope_Flat         279.659914       0.0  Reject Null (Keep Feature)
ChestPainType_ASY     243.021138       0.0  Reject Null (Keep Feature)
ExerciseAngina_N      222.259383       0.0  Reject Null (Keep Feature)
ExerciseAngina_Y      222.259383       0.0  Reject Null (Keep Feature)
ChestPainType_ATA     146.236323       0.0  Reject Null (Keep Feature)
is_Male                84.145101       0.0  Reject Null (Keep Feature)
FastingBS              64.320679       0.0  Reject Null (Keep Feature)
```

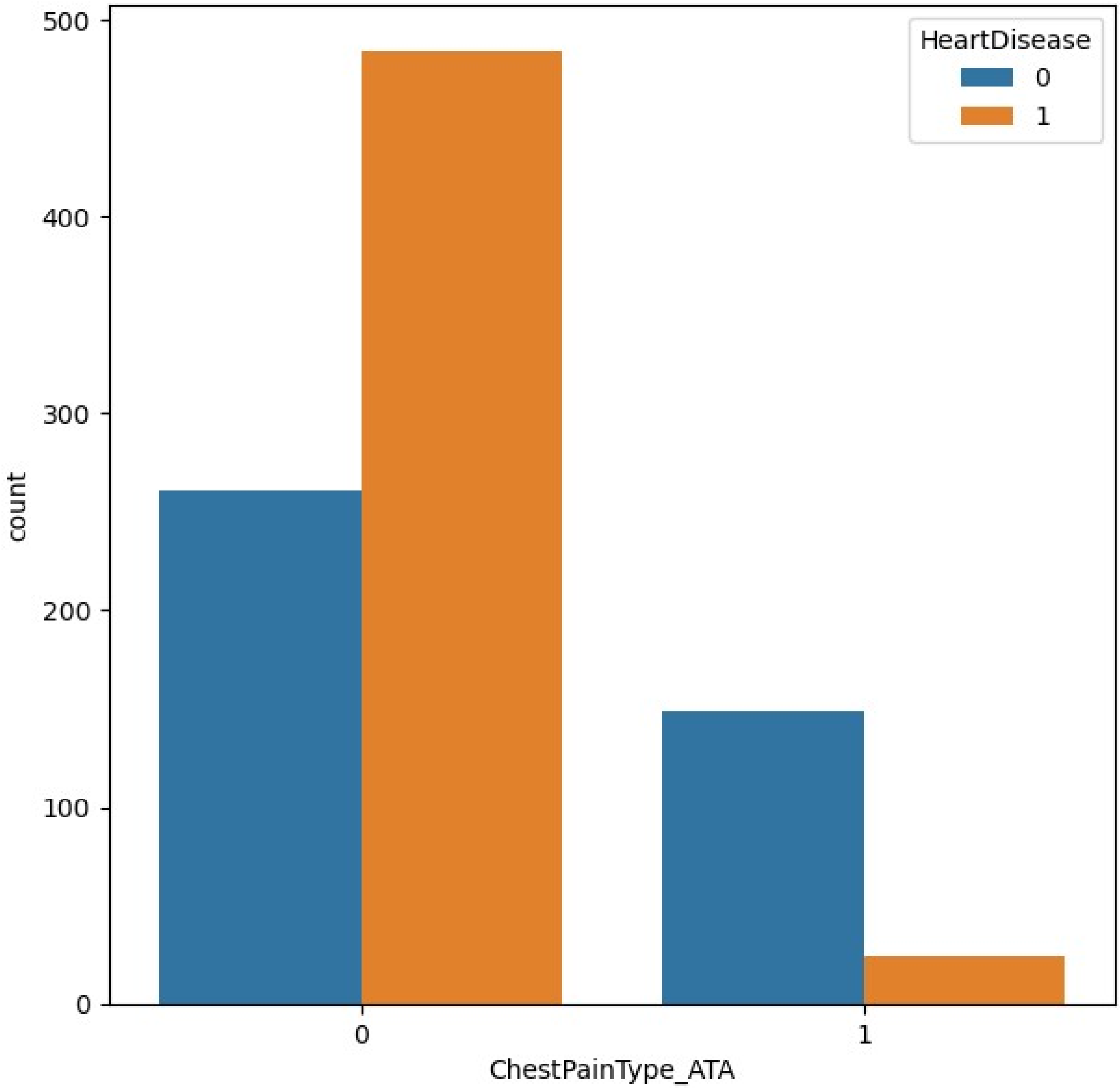
```
ChestPainType_NAP      40.608711       0.0  Reject Null (Keep Feature)
ST_Slope_Down          12.824125  0.000342  Reject Null (Keep Feature)
RestingECG_ST           9.135266  0.002507  Reject Null (Keep Feature)
RestingECG_Normal       7.327532  0.006791  Reject Null (Keep Feature)
ChestPainType_TA        2.273802  0.131577  Accept Null (Drop Feature)
RestingECG_LVH          0.058098  0.809528  Accept Null (Drop Feature)
```

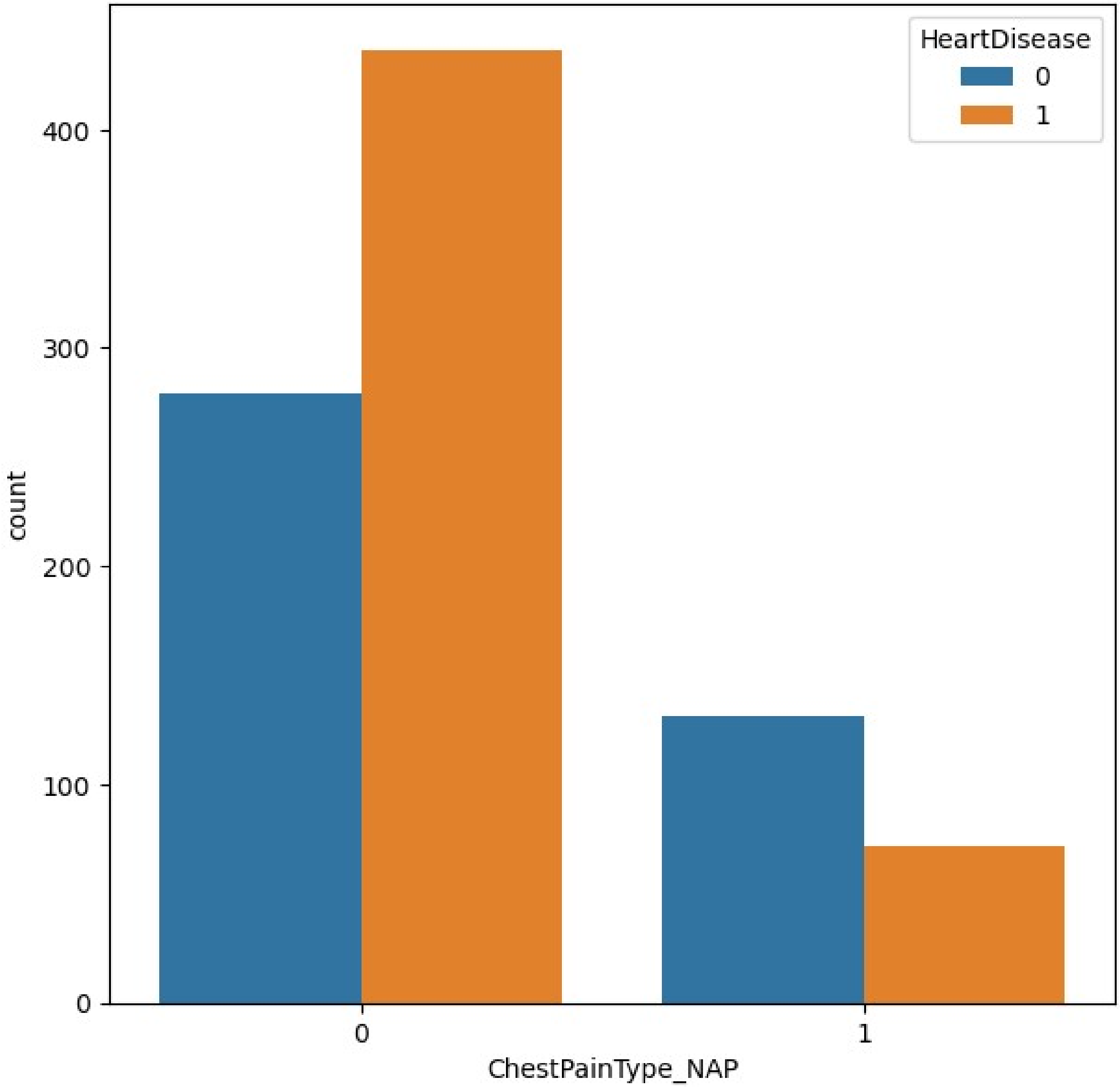
```
for i in cat_features:
    plt.figure(figsize=(7,7))
    sns.countplot(x=df_clean[i],hue=df_clean['HeartDisease'])
```

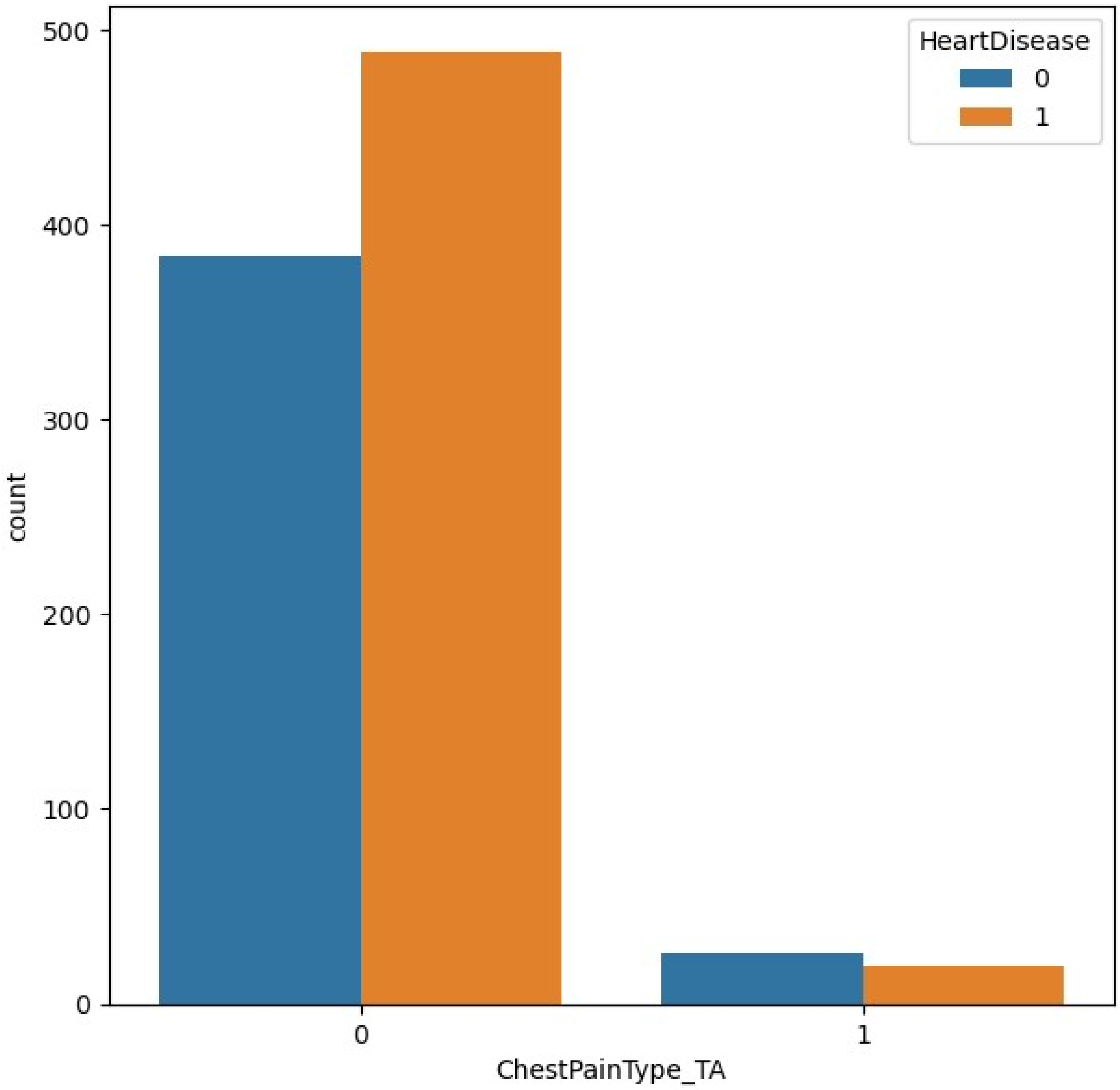


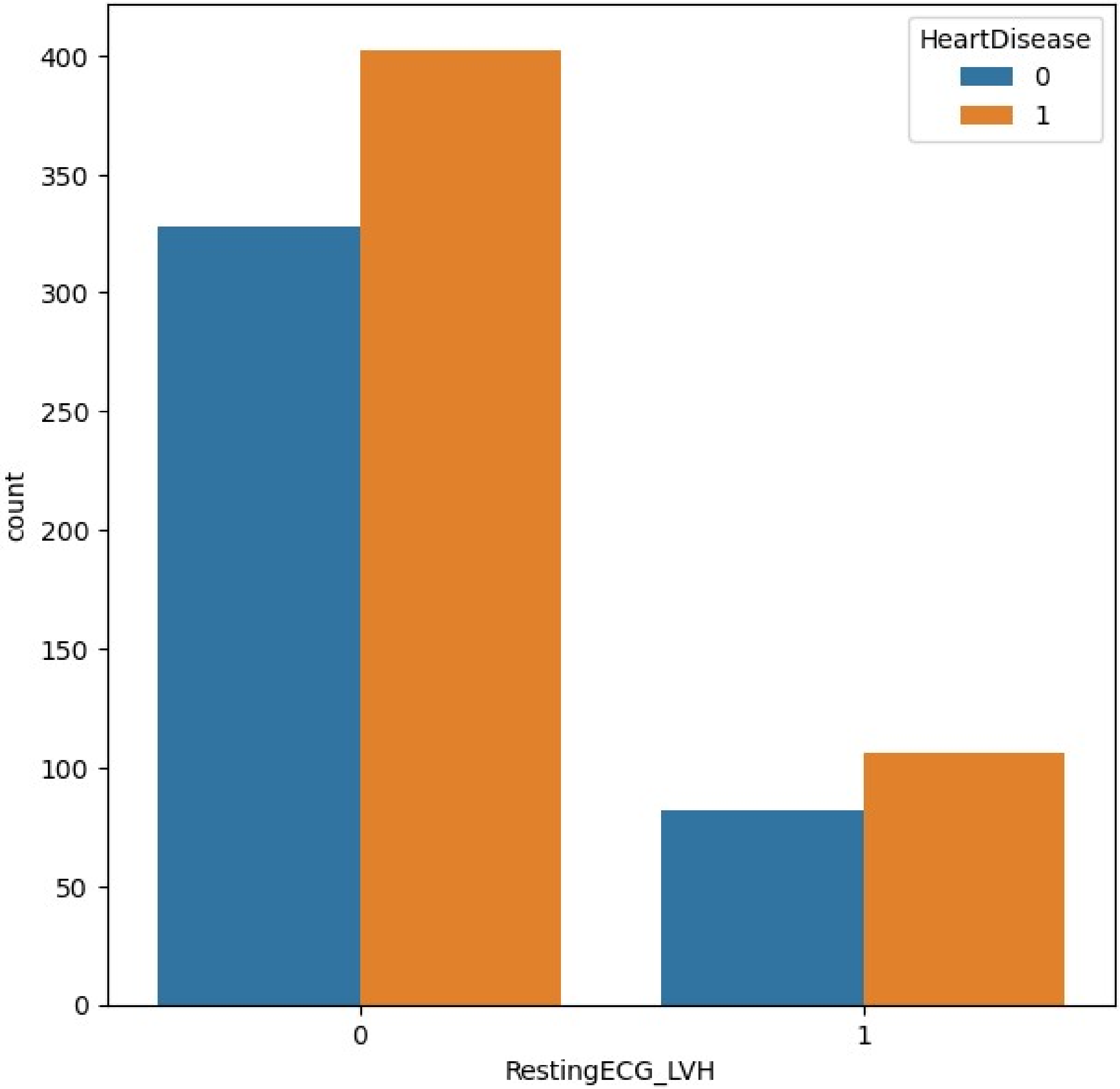

HeartDisease
🟦 0
🟧 1
count
0
50
100
150
200
250
300
350
0
1
FastingBS

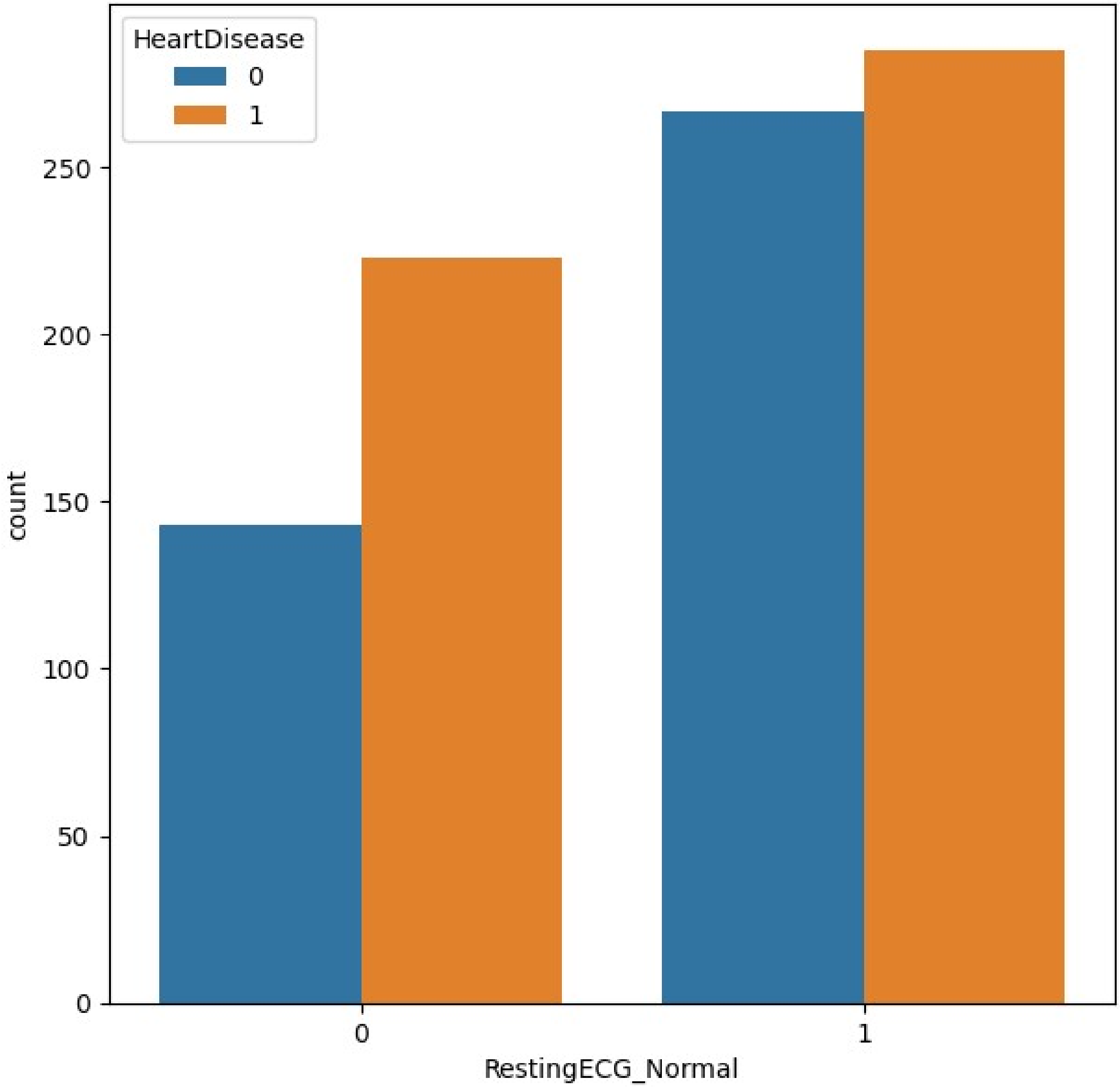

HeartDisease
0
1
400
350
300
250
200
150
100
50
0
count
0
1
ChestPainType_ASY

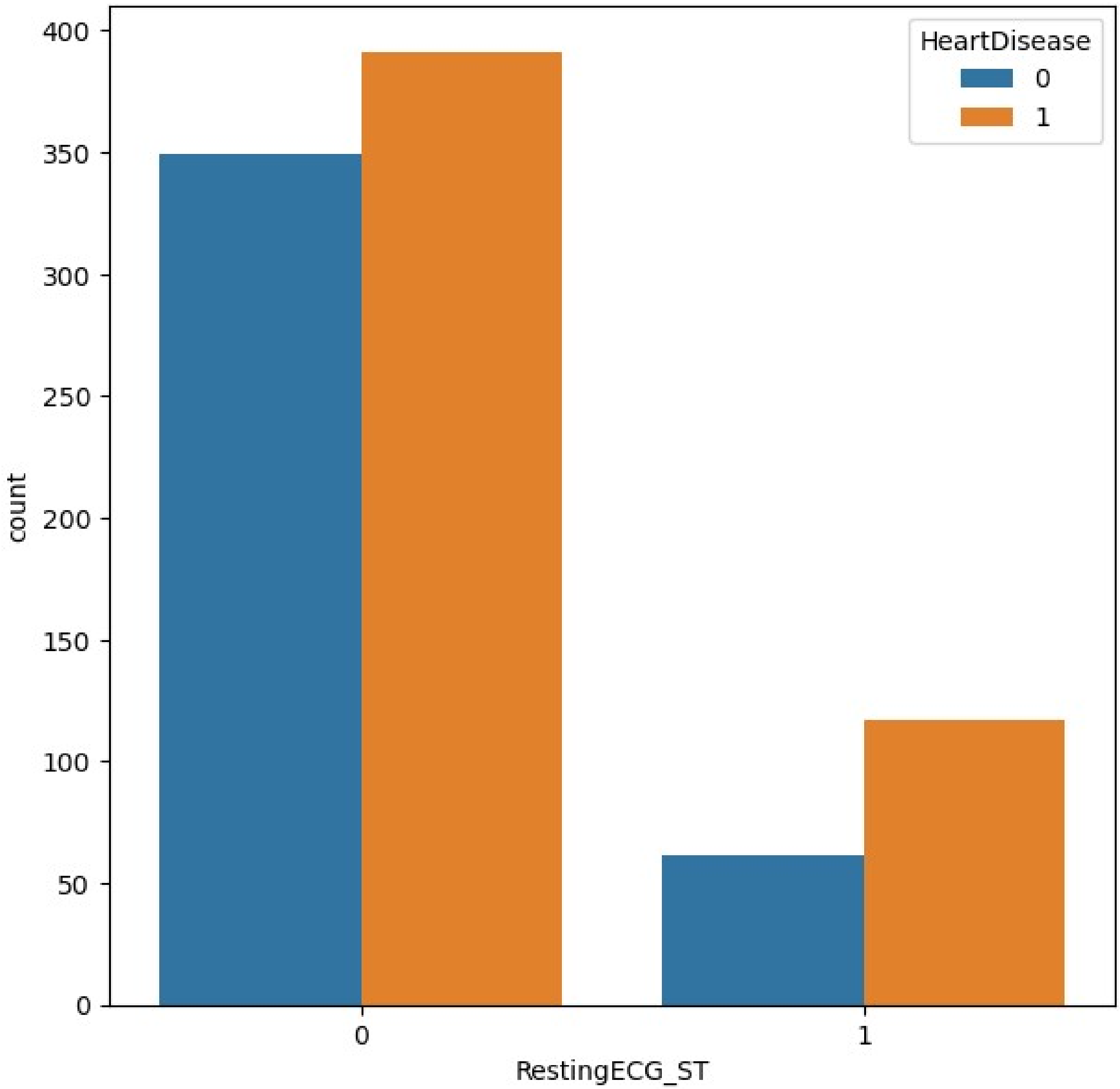

HeartDisease
0
1
count
500
400
300
200
100
0
0
1
ChestPainType_ATA

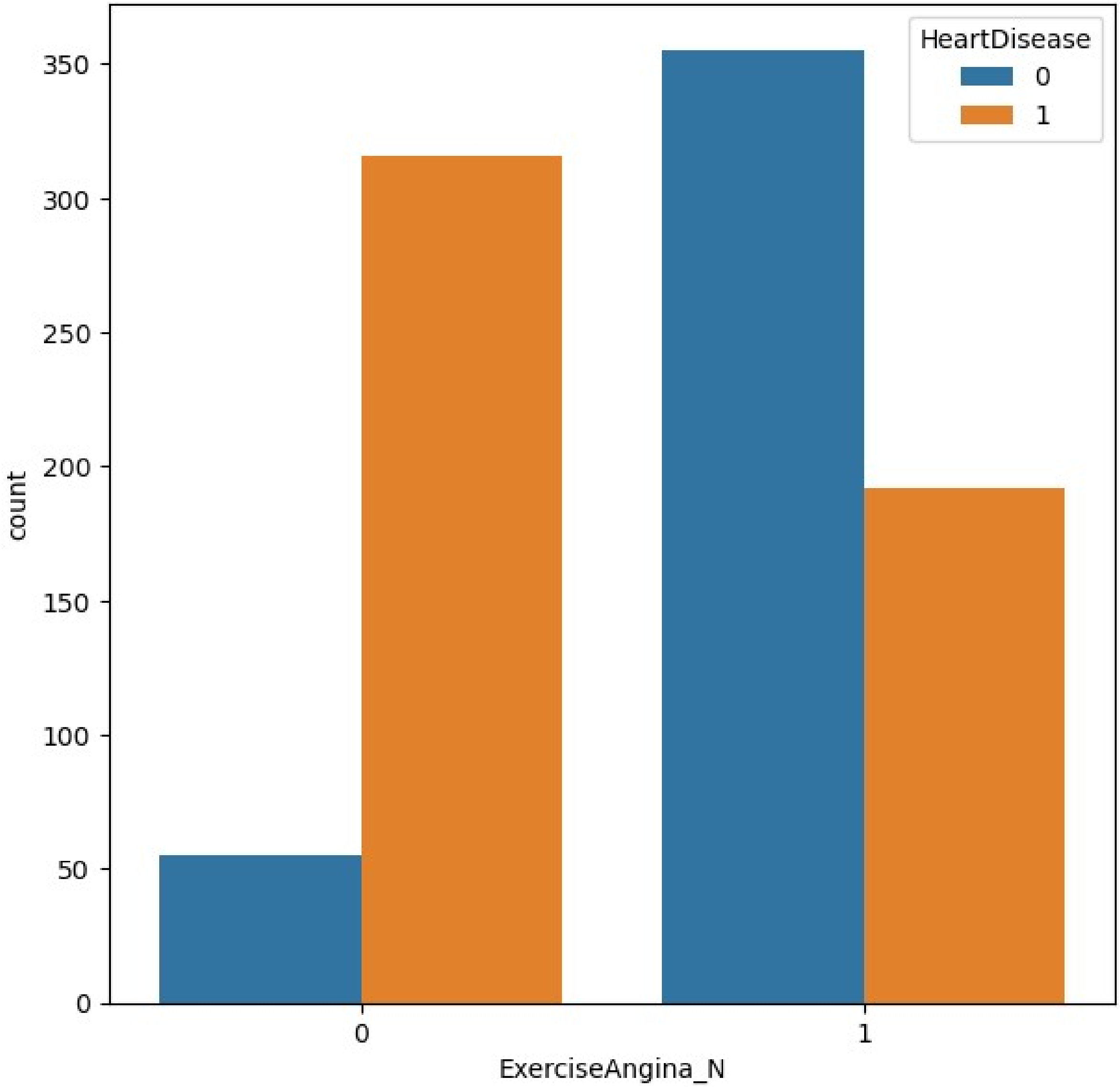

HeartDisease
0
1
count
0
100
200
300
400
0
1
ChestPainType_NAP

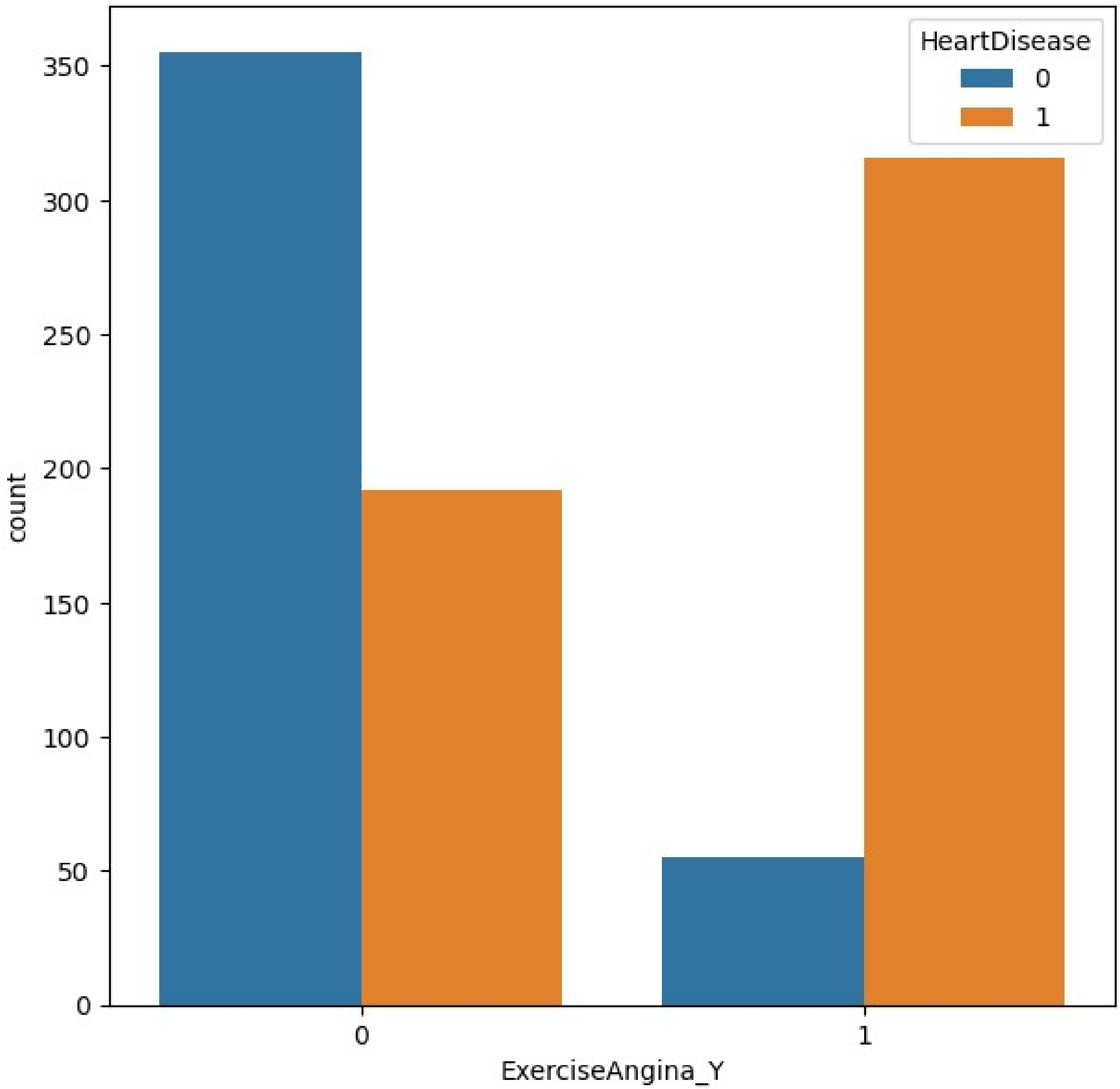

HeartDisease
🟦 0
🟧 1
count
500
400
300
200
100
0
0
1
ChestPainType_TA

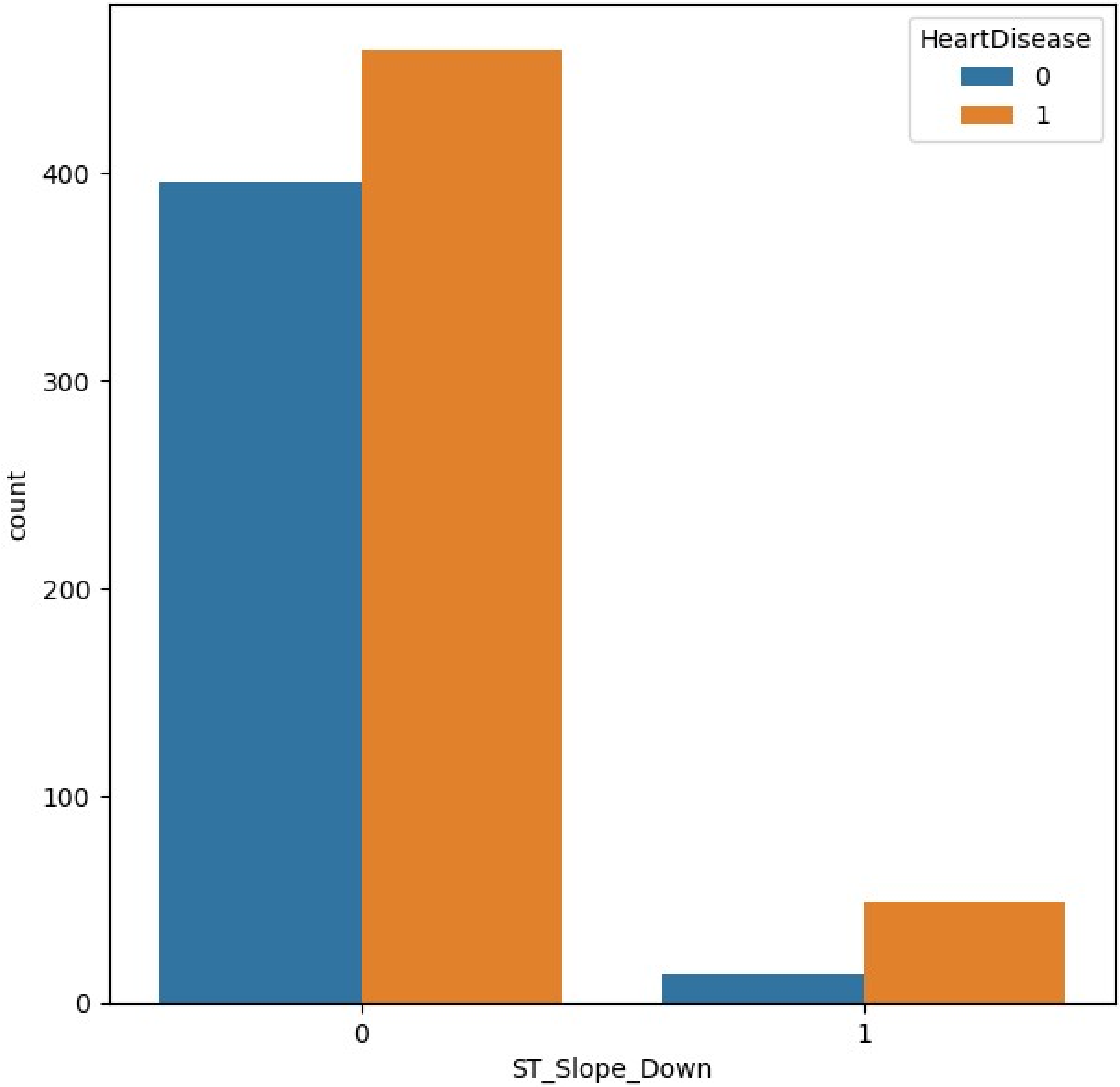

HeartDisease
0
1
count
400
350
300
250
200
150
100
50
0
0
1
RestingECG_LVH

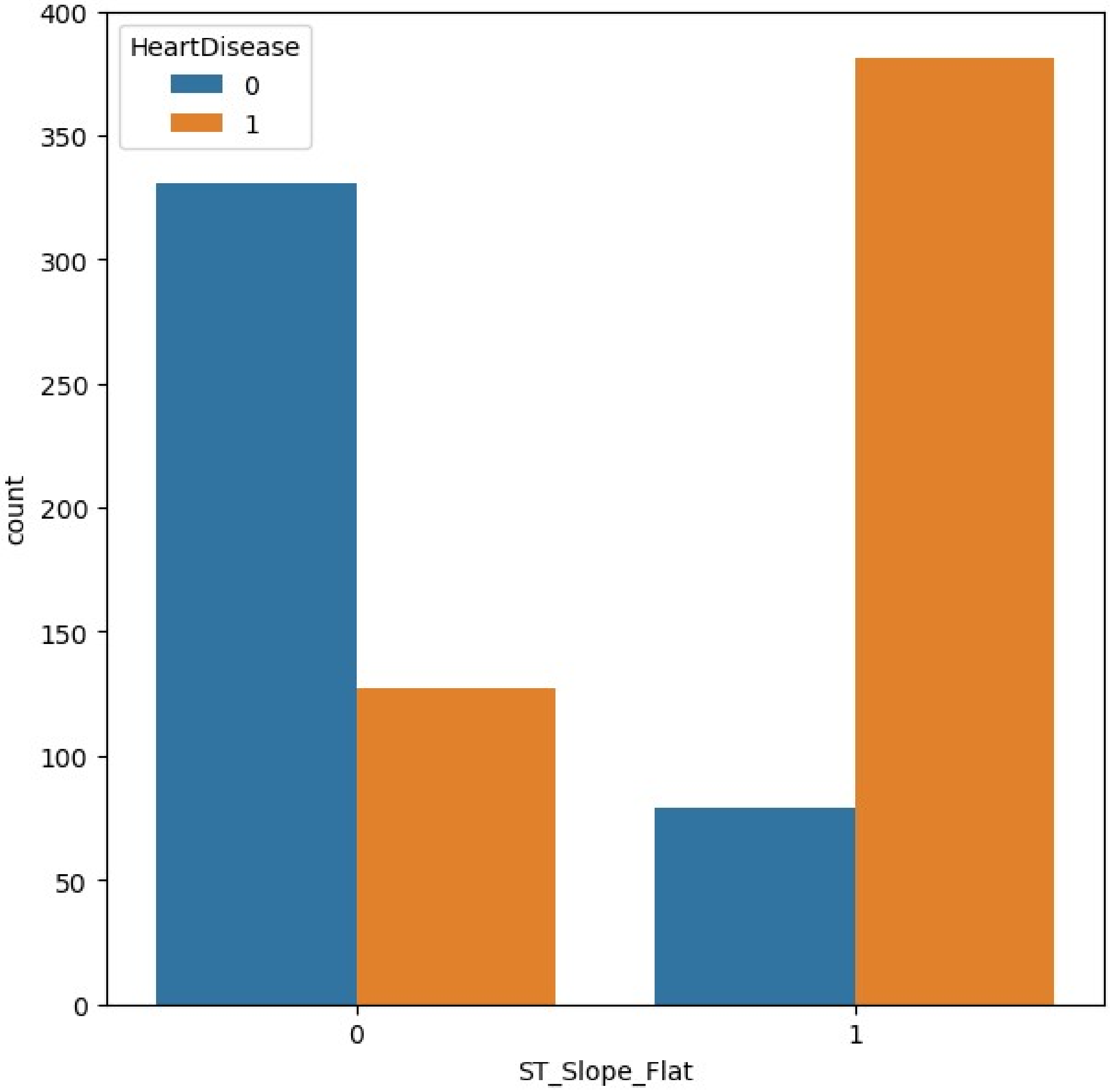

HeartDisease
0
1
count
250
200
150
100
50
0
0
1
RestingECG_Normal

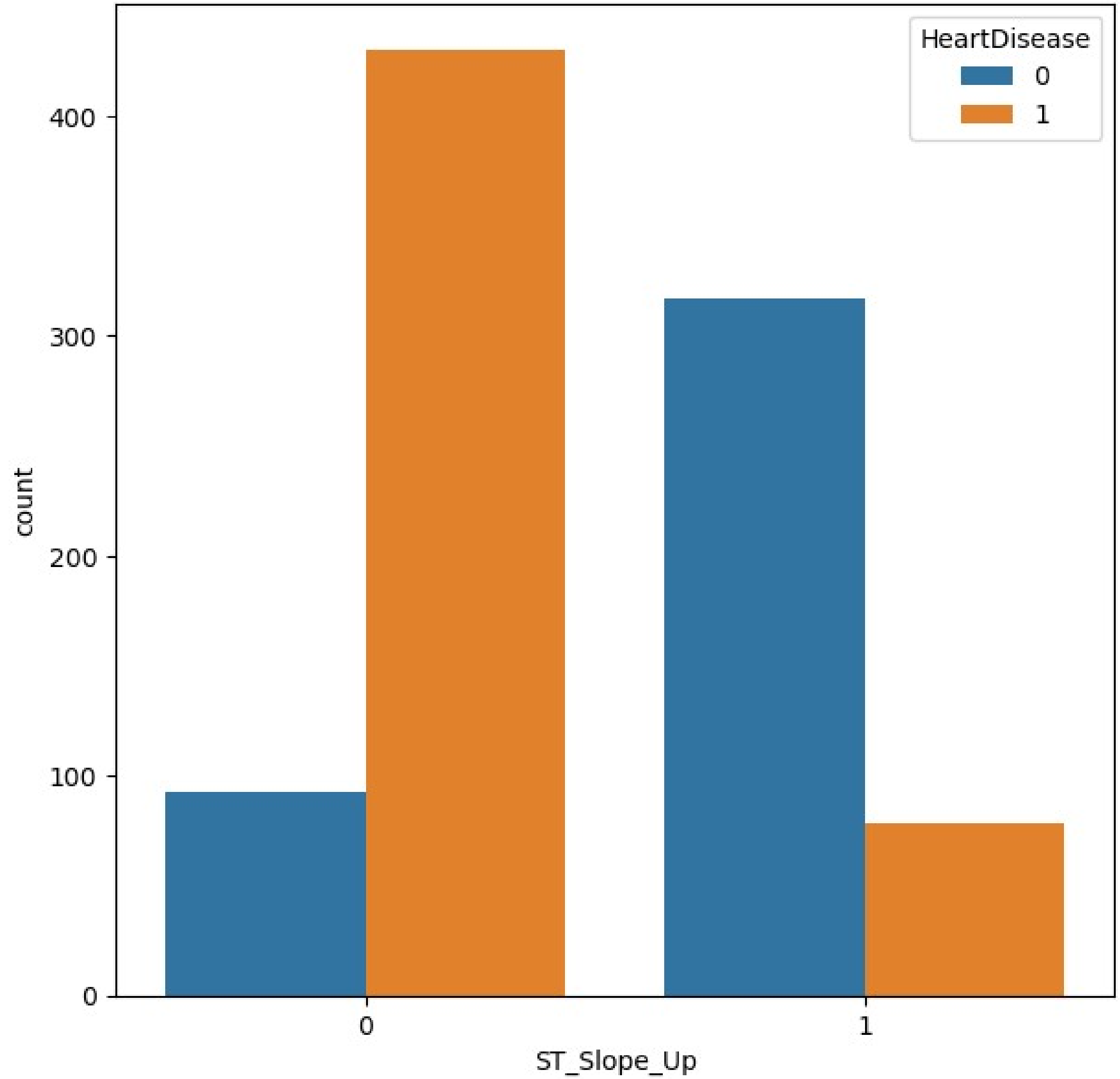

HeartDisease
0
1
count
400
350
300
250
200
150
100
50
0
RestingECG_ST


HeartDisease
0
1
count
350
300
250
200
150
100
50
0
0
1
ExerciseAngina_N


HeartDisease
0
1
count
350
300
250
200
150
100
50
0
0
1
ExerciseAngina_Y


HeartDisease
0
1
400
300
200
100
0
count
0
1
ST_Slope_Down


HeartDisease
0
1
400
350
300
250
200
150
100
50
0
count
0
1
ST_Slope_Flat



```
df_clean.columns

Index(['Age', 'is_Male', 'RestingBP', 'Cholesterol', 'FastingBS', 
'MaxHR',
       'Oldpeak', 'HeartDisease', 'ChestPainType_ASY', 
'ChestPainType_ATA',
       'ChestPainType_NAP', 'ChestPainType_TA', 'RestingECG_LVH',
       'RestingECG_Normal', 'RestingECG_ST', 'ExerciseAngina_N',
       'ExerciseAngina_Y', 'ST_Slope_Down', 'ST_Slope_Flat', 
'ST_Slope_Up',
       'HeartDisease_bin'],
      dtype='object')

df_clean.head(5)
```

```
   Age  is_Male  RestingBP  Cholesterol  FastingBS  MaxHR  Oldpeak  \
0   40        1        140          289          0    172        0
1   49        0        160          180          0    156        1
2   37        1        130          283          0     98        0
3   48        0        138          214          0    108        1
4   54        1        150          195          0    122        0

   HeartDisease  ChestPainType_ASY  ChestPainType_ATA  ...
ChestPainType_TA  \
0             0                  0                  1  ...
0
1             1                  0                  0  ...
0
2             0                  0                  1  ...
0
3             1                  1                  0  ...
0
4             0                  0                  0  ...
0

   RestingECG_LVH  RestingECG_Normal  RestingECG_ST  ExerciseAngina_N
\
0               0                  1              0                 1

1               0                  1              0                 1

2               0                  0              1                 1

3               0                  1              0                 0

4               0                  1              0                 1


   ExerciseAngina_Y  ST_Slope_Down  ST_Slope_Flat  ST_Slope_Up  \
0                 0              0              0            1
1                 0              0              1            0
2                 0              0              0            1
3                 1              0              1            0
4                 0              0              0            1

   HeartDisease_bin
0                 0
1                 1
2                 0
3                 1
4                 0

[5 rows x 21 columns]
```

model

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, f1_score,
classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

df_clean.columns
```

```
Index(['Age', 'is_Male', 'RestingBP', 'Cholesterol', 'FastingBS',
'MaxHR',
       'Oldpeak', 'HeartDisease', 'ChestPainType_ASY',
'ChestPainType_ATA',
       'ChestPainType_NAP', 'ChestPainType_TA', 'RestingECG_LVH',
       'RestingECG_Normal', 'RestingECG_ST', 'ExerciseAngina_N',
       'ExerciseAngina_Y', 'ST_Slope_Down', 'ST_Slope_Flat',
'ST_Slope_Up',
       'HeartDisease_bin'],
      dtype='object')
```

```
X = df_clean.drop(columns=[ 'HeartDisease', 'ExerciseAngina_N',
       'HeartDisease_bin'], axis=1)
y = df_clean['HeartDisease']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, stratify=y, test_size=0.2, random_state=42)

scaler=StandardScaler()
x_train_scaled=scaler.fit_transform(X_train)
x_test_scaled=scaler.fit_transform(X_test)

models = {
    "Logistic Regression": LogisticRegression(),
    "KNN": KNeighborsClassifier(n_neighbors=7),
    "Naive Bayes": GaussianNB(),
    "Decision Tree": DecisionTreeClassifier(),
    "SVM (RBF Kernel)": SVC(probability=True,kernel='poly')
}

results = []

for name, model in models.items():
    model.fit(x_train_scaled, y_train)
    y_pred = model.predict(x_test_scaled)
    acc = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    results.append({
        'Model': name,
        'Accuracy': round(acc, 4),
        'F1 Score': round(f1, 4)
    })

results
```

```
[{'Model': 'Logistic Regression', 'Accuracy': 0.8967, 'F1 Score':
0.91},
 {'Model': 'KNN', 'Accuracy': 0.875, 'F1 Score': 0.89},
 {'Model': 'Naive Bayes', 'Accuracy': 0.8587, 'F1 Score': 0.8725},
 {'Model': 'Decision Tree', 'Accuracy': 0.7554, 'F1 Score': 0.7739},
 {'Model': 'SVM (RBF Kernel)', 'Accuracy': 0.875, 'F1 Score': 0.891}]
```

```
import joblib
joblib.dump(models['Logistic Regression'],'LOG_heart.pkl')
joblib.dump(scaler,'scaler.pkl')
joblib.dump(X.columns.tolist(),'columns.pkl')
```

```
['columns.pkl']
```

```
X.columns
```

```
Index(['Age', 'is_Male', 'RestingBP', 'Cholesterol', 'FastingBS',
'MaxHR',
       'Oldpeak', 'ChestPainType_ASY', 'ChestPainType_ATA',
       'ChestPainType_NAP', 'ChestPainType_TA', 'RestingECG_LVH',
       'RestingECG_Normal', 'RestingECG_ST', 'ExerciseAngina_Y',
       'ST_Slope_Down', 'ST_Slope_Flat', 'ST_Slope_Up'],
      dtype='object')
```