```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import scale



data = pd.read_csv('heart_disease.csv')
data.dropna()
print(data.head(10))
```

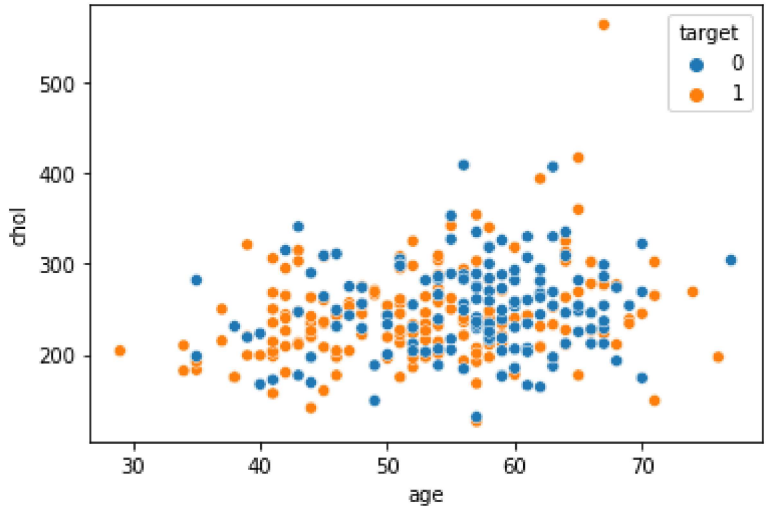
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   3       145   233    1        0      150      0      2.3      0
1   37    1   2       130   250    0        1      187      0      3.5      0
2   41    0   1       130   204    0        0      172      0      1.4      2
3   56    1   1       120   236    0        1      178      0      0.8      2
4   57    0   0       120   354    0        1      163      1      0.6      2
5   57    1   0       140   192    0        1      148      0      0.4      1
6   56    0   1       140   294    0        0      153      0      1.3      1
7   44    1   1       120   263    0        1      173      0      0.0      2
8   52    1   2       172   199    1        1      162      0      0.5      2
9   57    1   2       150   168    0        1      174      0      1.6      2

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
5   0     1       1
6   0     2       1
7   0     3       1
8   0     3       1
9   0     2       1
```

```
sns.scatterplot(x="age", y="chol", data=data.head(100))
```

```
<AxesSubplot:xlabel='age', ylabel='chol'>
```

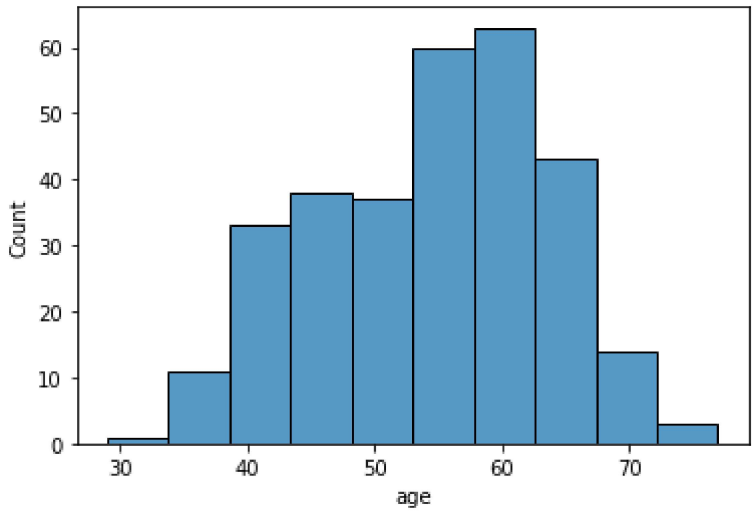

```
sns.scatterplot(x="age",
                    y="chol",
                    hue="target",
                    data=data)
```

```
<AxesSubplot:xlabel='age', ylabel='chol'>
```



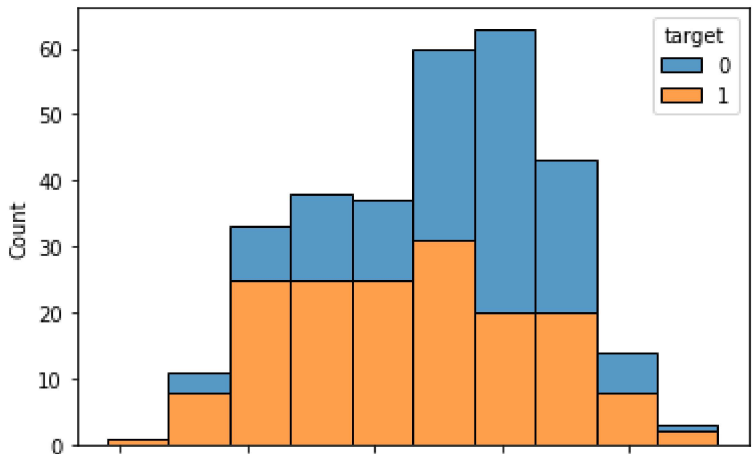

```
sns.histplot(x = 'age', data=data, bins=10)
```

```
<AxesSubplot:xlabel='age', ylabel='Count'>
```



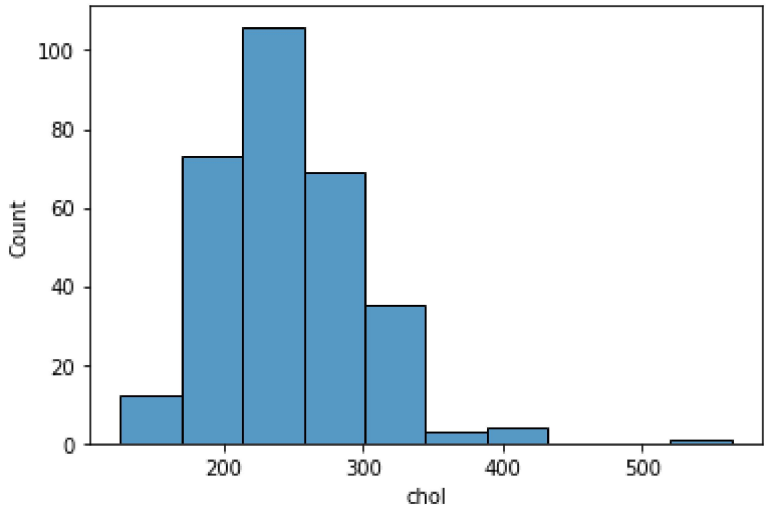

```
sns.histplot(x = 'age', hue='target', data=data, bins=10, multiple='stack')
```

```
<AxesSubplot:xlabel='age', ylabel='Count'>
```




```
sns.histplot(x = 'chol', data=data, bins=10)
```

```
<AxesSubplot:xlabel='chol', ylabel='Count'>
```




```
features=['age','chol','cp']
target='target'
x= data[features].values
y= data[target].values
print(x, y)
```

```
[ 45 309   0]
[ 58 259   0]
[ 50 200   0]
[ 62 244   0]
[ 38 231   3]
[ 66 228   0]
[ 52 230   0]
[ 53 282   0]
[ 63 269   0]
[ 54 206   0]
[ 66 212   0]
[ 55 327   0]
[ 49 149   2]
[ 54 286   0]
[ 56 283   0]
```

```
 [ 46 249   0]
 [ 61 234   3]
 [ 67 237   0]
 [ 58 234   0]
 [ 47 275   0]
 [ 52 212   0]
 [ 58 218   0]
 [ 57 261   1]
 [ 58 319   1]
 [ 61 166   0]
 [ 42 315   0]
 [ 52 204   0]
 [ 59 218   2]
 [ 40 223   0]
 [ 61 207   0]
 [ 46 311   0]
 [ 59 204   3]
 [ 57 232   1]
 [ 57 335   0]
 [ 55 205   0]
 [ 61 203   0]
 [ 58 318   0]
 [ 58 225   0]
 [ 67 212   2]
 [ 44 169   0]
 [ 63 187   0]
 [ 63 197   0]
 [ 59 176   0]
 [ 57 241   0]
 [ 45 264   3]
 [ 68 193   0]
 [ 57 131   0]
 [ 57 236   1]] [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0]
```

```
x_train,x_test,y_train,y_test =  train_test_split (x,y,test_size=0.3,random_state=10)
```

```
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train, y_train)
```

```
KNeighborsClassifier()
```

```
#train accuracy
y_pred = classifier.predict(x_train)
```

```
print("training accuracy: ",accuracy_score(y_train, y_pred))
#test accuracy
y_pred = classifier.predict(x_test)
print("testing accuracy: ",accuracy_score(y_test, y_pred))
```

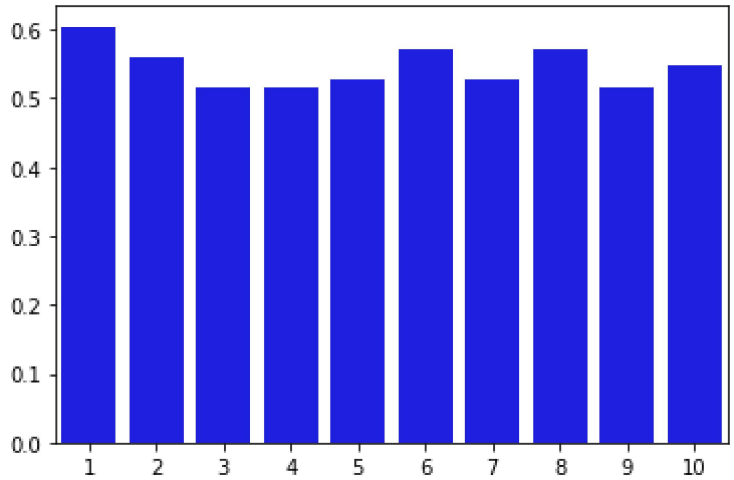
```
training accuracy:  0.7028301886792453
testing accuracy:  0.5274725274725275
```

```
k_values = []
training_acc_Scores = []
testing_acc_Scores = []
for i in range(1, 11):
    classifier = KNeighborsClassifier(n_neighbors=i)
    classifier.fit(x_train, y_train)
    y_pred = classifier.predict(x_train)
    training_acc_Scores.append(accuracy_score(y_train, y_pred))
    y_pred = classifier.predict(x_test)
    testing_acc_Scores.append(accuracy_score(y_test, y_pred))
    k_values.append(i)


print(k_values)
print(training_acc_Scores)
print(testing_acc_Scores)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[0.9952830188679245, 0.7783018867924528, 0.7641509433962265, 0.7594339622641509, 0.7028
[0.6043956043956044, 0.5604395604395604, 0.5164835164835165, 0.5164835164835165, 0.5274
```

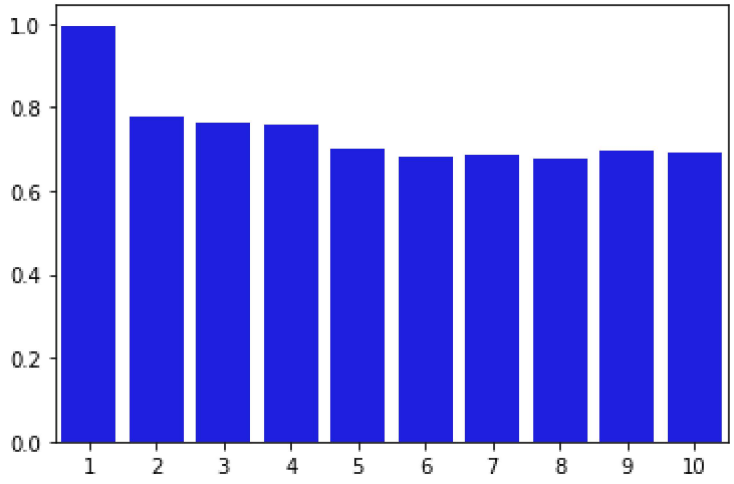
```
sns.barplot(x=k_values,y=testing_acc_Scores,color='blue')
```

```
<AxesSubplot:>
```




```
sns.barplot(x=k_values,y=training_acc_Scores,color='blue')
```

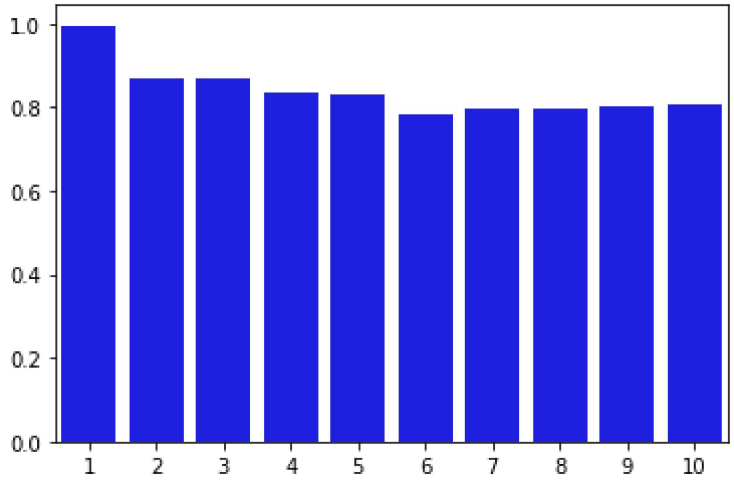
```
<AxesSubplot:>
```




```
# with preprocessing
x_scaled_train = scale(x_train)
k_values = []
training_acc_Scores = []
testing_acc_Scores = []
for i in range(1, 11):
    classifier = KNeighborsClassifier(n_neighbors=i)
    classifier.fit(x_scaled_train, y_train)
    y_pred = classifier.predict(x_scaled_train)
    training_acc_Scores.append(accuracy_score(y_train, y_pred))
    y_pred = classifier.predict(x_test)
    testing_acc_Scores.append(accuracy_score(y_test, y_pred))
    k_values.append(i)
```

```
sns.barplot(x=k_values,y=training_acc_Scores,color='blue')
```

```
<AxesSubplot:>
```




```
sns.barplot(x=k_values,y=testing_acc_Scores,color='blue')
```

```
<AxesSubplot:>
```