Visualization

December 5, 2019

[1]:

```
import numpy as np
import pandas as pd
from sklearn import tree, metrics
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix
from scipy import misc
import collections
from matplotlib import pyplot as plt
import seaborn as sns
```

[2]:

```
data = pd.read_csv('heart.csv')
```

[3]:

```
data.head()
```

[3]:

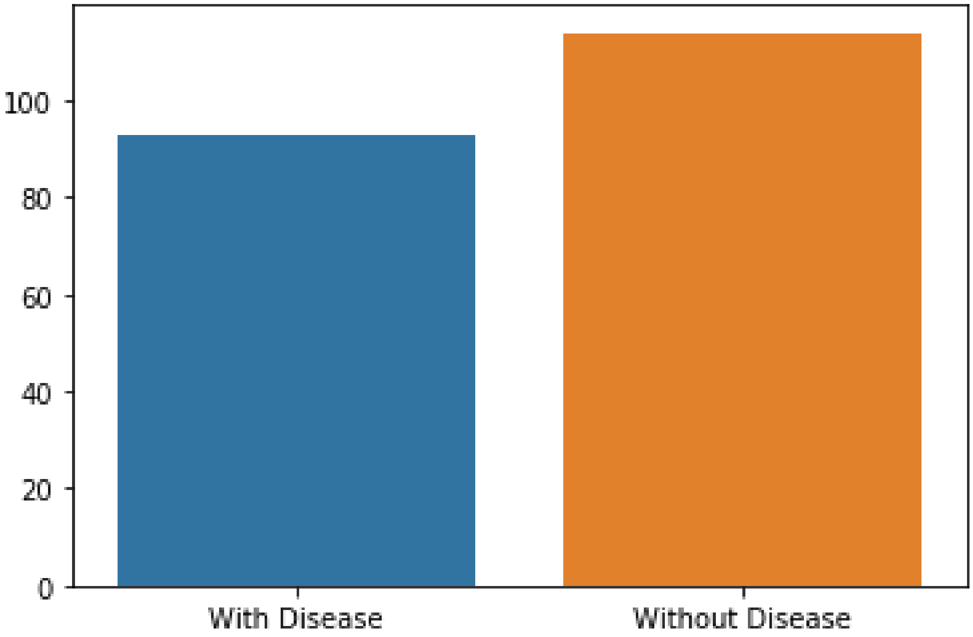
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   3       145   233    1        0      150      0      2.3      0
1   37    1   2       130   250    0        1      187      0      3.5      0
2   41    0   1       130   204    0        0      172      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

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
```

[4]:

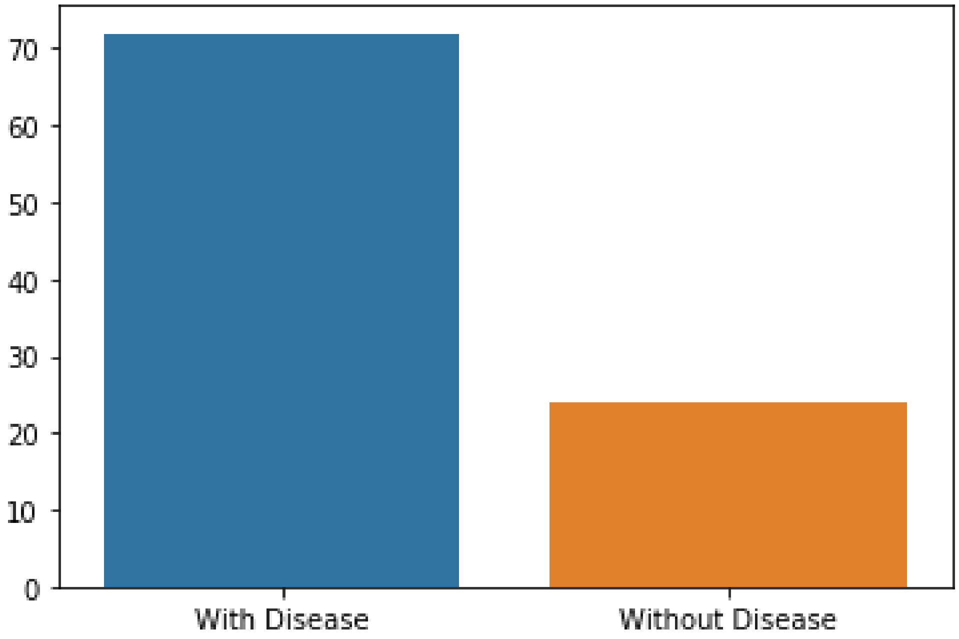
```
def feature_target_relationship(fname, val):
    with_disease=len(data[(data[fname] == val) & (data['target'] == 1)])
    without_disease=len(data[(data[fname] == val) & (data['target'] == 0)])
    sns.barplot(x=['With Disease','Without Disease'], y=[with_disease,␣
 ↪without_disease])
    plt.show()

def print_info(fname, val):
    print("---------------------------------------------------------")
    print("Feature: " + fname)
    print("Value: ", val)
```

[5]:
```
print_info("sex", "male")
feature_target_relationship("sex", 1)
print_info("sex", "female")
feature_target_relationship("sex", 0)
```

```
---------------------------------------------------------
Feature: sex
Value:  male
```



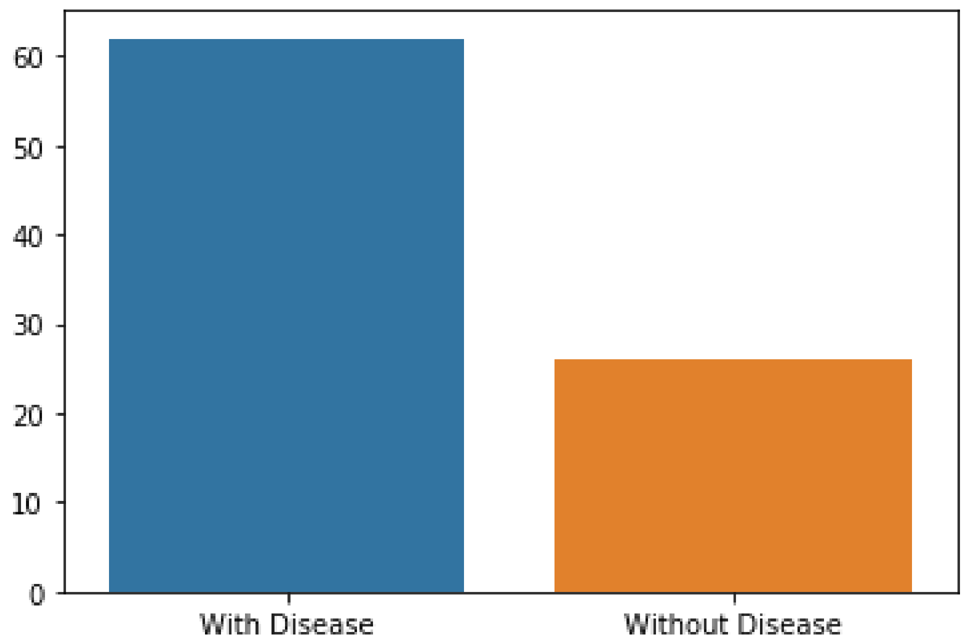
```
---------------------------------------------------------
Feature: sex
Value:  female
```



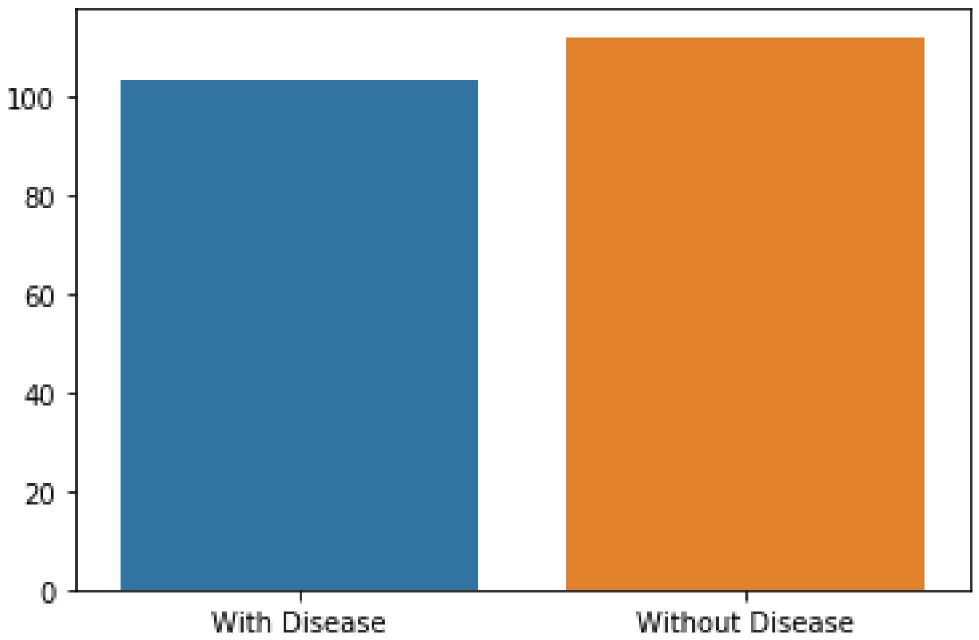
[6]:

```
print_info('Age', 'Young people')
with_disease=len(data[(data['age'] < 50) & (data['target']==1)])
without_disease=len(data[(data['age'] < 50) & (data['target']==0)])
sns.barplot(x=['With Disease','Without Disease'], y=[with_disease,␣
 ↪without_disease])
plt.show()
print_info('Age', 'Elders')
with_disease=len(data[(data['age'] >= 50) & (data['target']==1)])
without_disease=len(data[(data['age'] >= 50) & (data['target']==0)])
sns.barplot(x=['With Disease','Without Disease'], y=[with_disease,␣
 ↪without_disease])
plt.show()
```

```
---------------------------------------------------------
Feature: Age
Value:  Young people
```

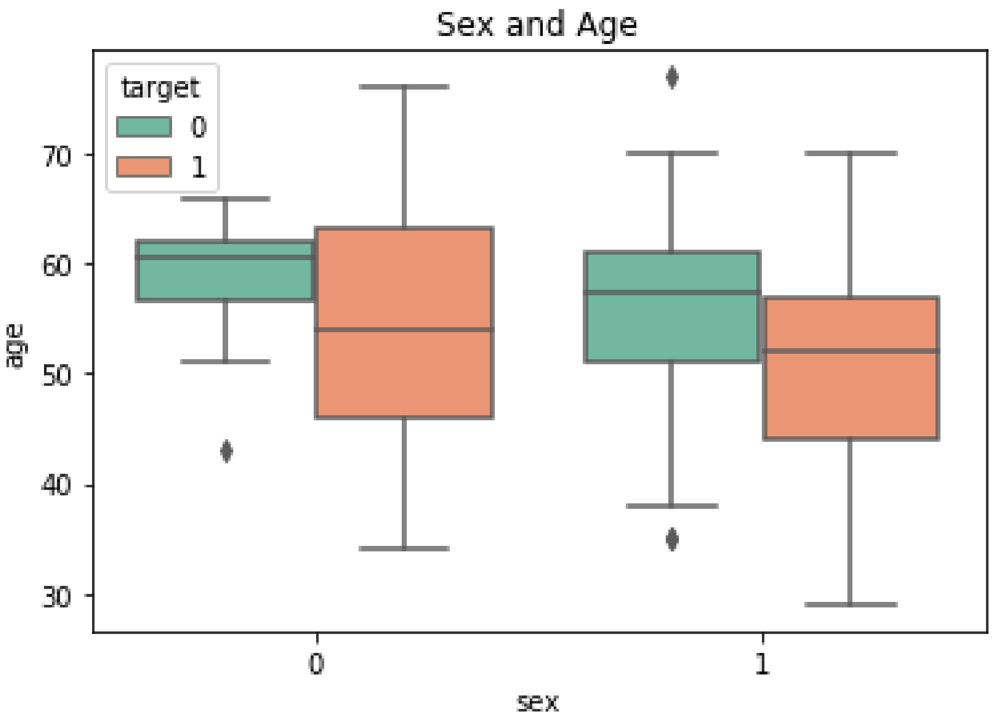


```
----------------------------------------------------------
Feature: Age
Value:  Elders
```



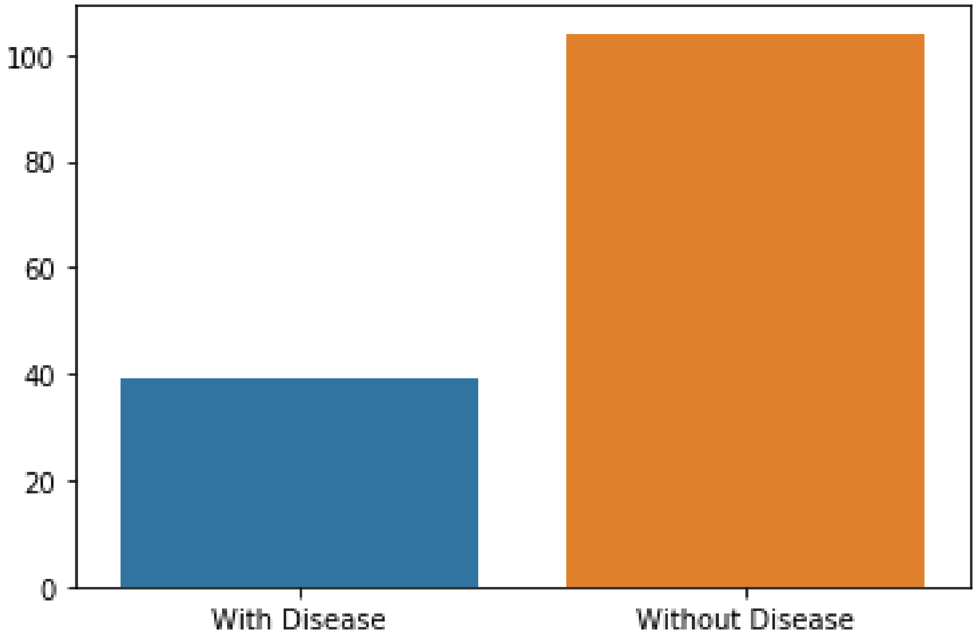
[7]:
```
sns.boxplot(x='sex', y='age', data=data, hue='target', palette='Set2')
plt.title("Sex and Age")
```

[7]:
```
Text(0.5, 1.0, 'Sex and Age')
```

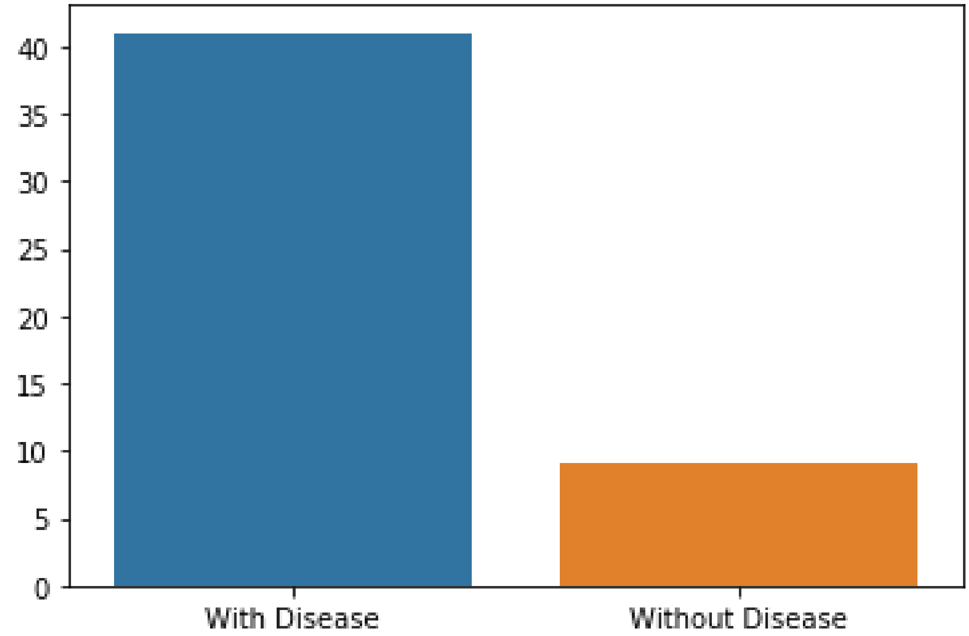


[8]:
```
fname = 'The chest pain experienced '
print_info(fname, "0")
feature_target_relationship('cp', 0)
print_info(fname, "1")
feature_target_relationship("cp", 1)
print_info(fname, 2)
feature_target_relationship("cp", 2)
print_info(fname, 3)
feature_target_relationship("cp", 3)
```

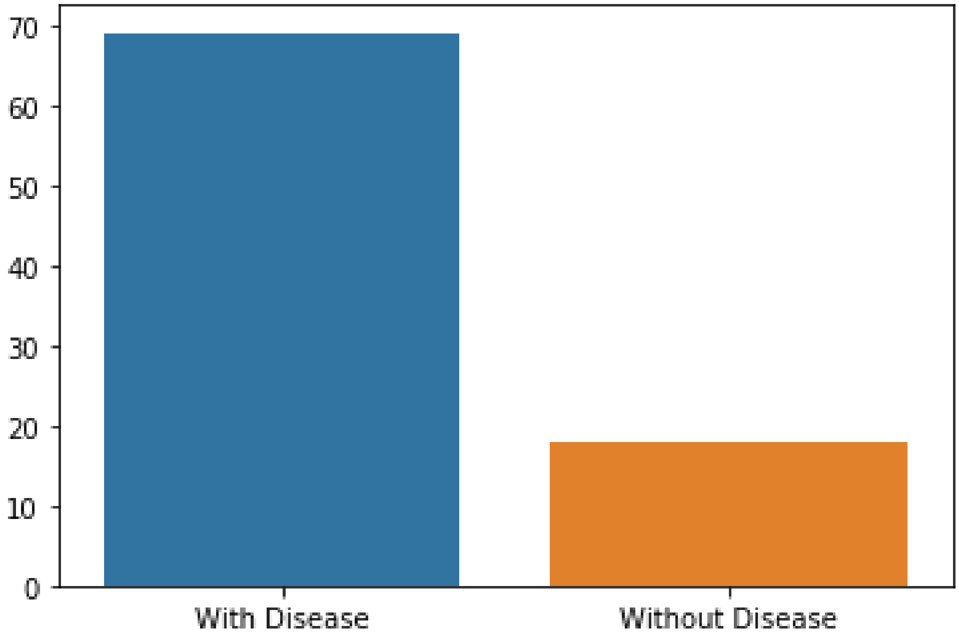
```
---------------------------------------------------------
Feature: The chest pain experienced
Value:  0
```



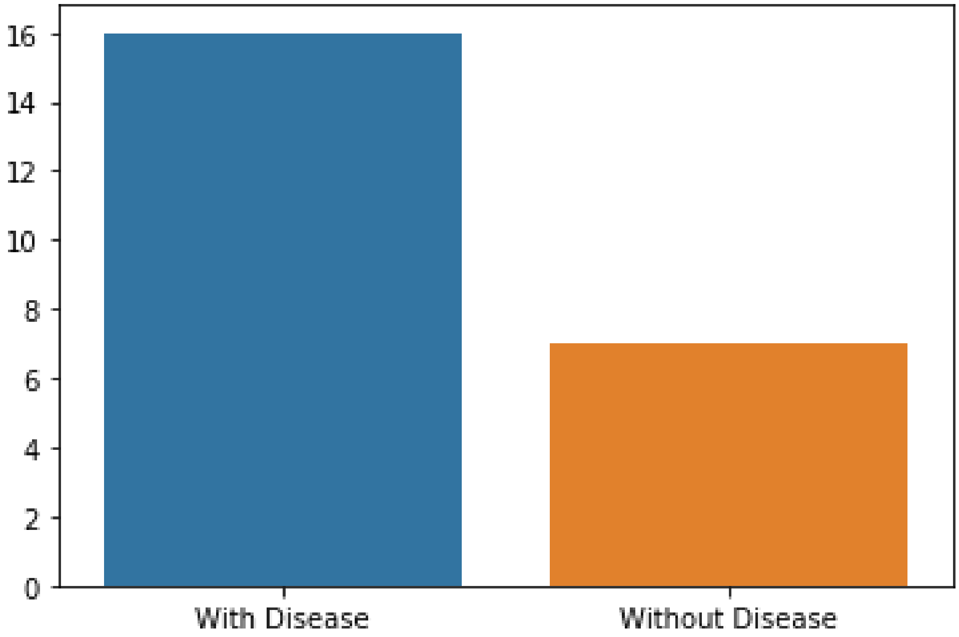
```
------------------------------------------------------------
Feature: The chest pain experienced
Value:  1
```



```
---------------------------------------------------------
Feature: The chest pain experienced
Value:  2
```



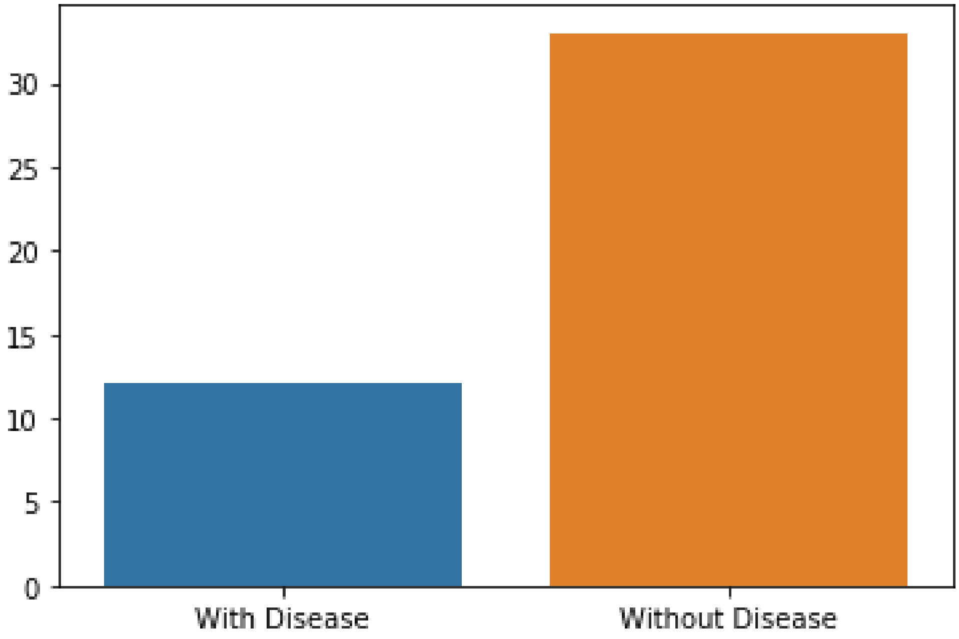
```
---------------------------------------------------------
Feature: The chest pain experienced
Value:  3
```



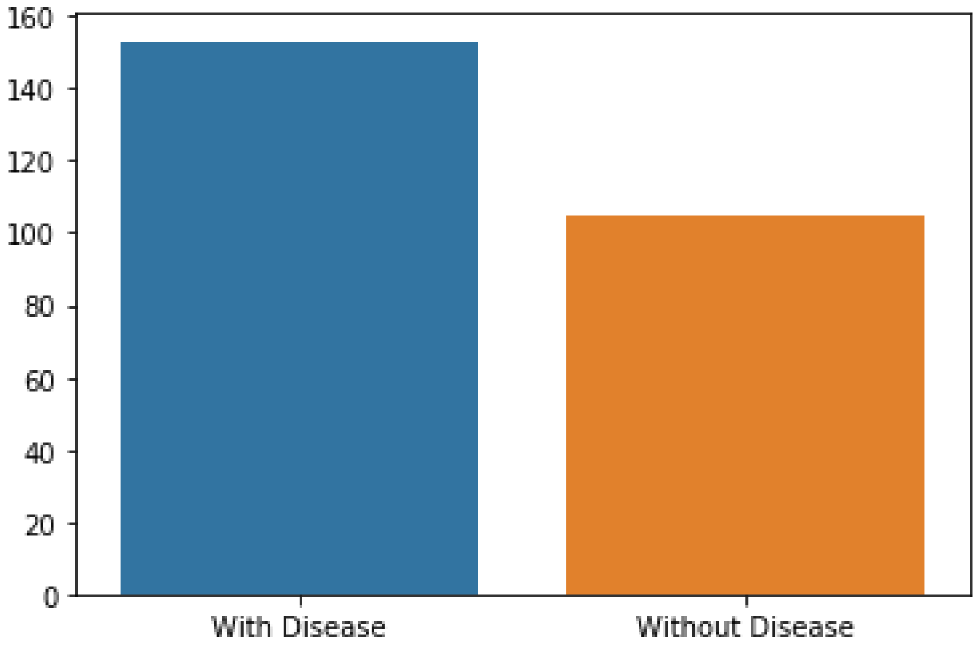
[9]:

```
fname = 'Maximum heart rate achieved'
print_info(fname, 'slow')
thred = 125
with_disease=len(data[(data['thalach'] < thred) & (data['target']==1)])
without_disease=len(data[(data['thalach'] < thred) & (data['target']==0)])
sns.barplot(x=['With Disease','Without Disease'], y=[with_disease,␣
 ↪without_disease])
plt.show()
print_info(fname, 'fast')
with_disease=len(data[(data['thalach'] >= thred) & (data['target']==1)])
without_disease=len(data[(data['thalach'] >= thred) & (data['target']==0)])
sns.barplot(x=['With Disease','Without Disease'], y=[with_disease,␣
 ↪without_disease])
plt.show()
```

```
--------------------------------------------------------------
Feature: Maximum heart rate achieved
Value:  slow
```



```
--------------------------------------------------------------
Feature: Maximum heart rate achieved
Value:  fast
```

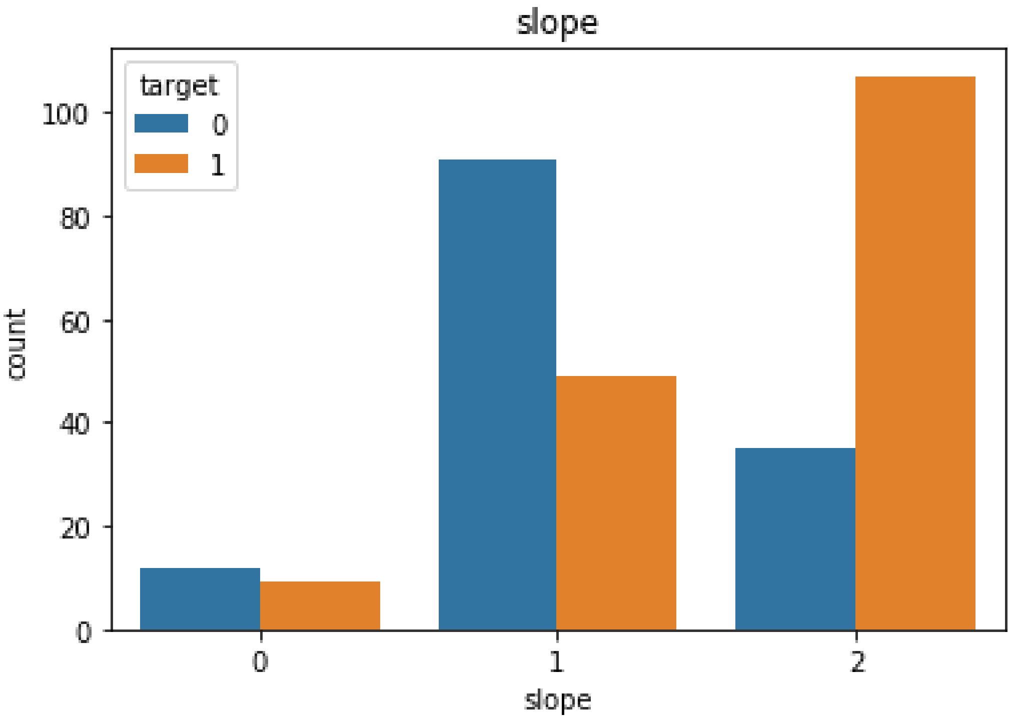


[10]:
```
data.head()
```

[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

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
```

[11]:
```
plot = sns.countplot(x='slope', data=data, hue='target')␣
 ↪#,palette='plasma',linewidth=3)
plot.set_title("slope")
```

[11]:
```
Text(0.5, 1.0, 'slope')
```


slope
target
0
1
100
80
60
40
20
0
count
0
1
2
slope

[ ]:

[ ]: