

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
In [3]: df=pd.read_csv('heart.csv')
```

```
In [4]: df.head()
```

Out[4]:

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

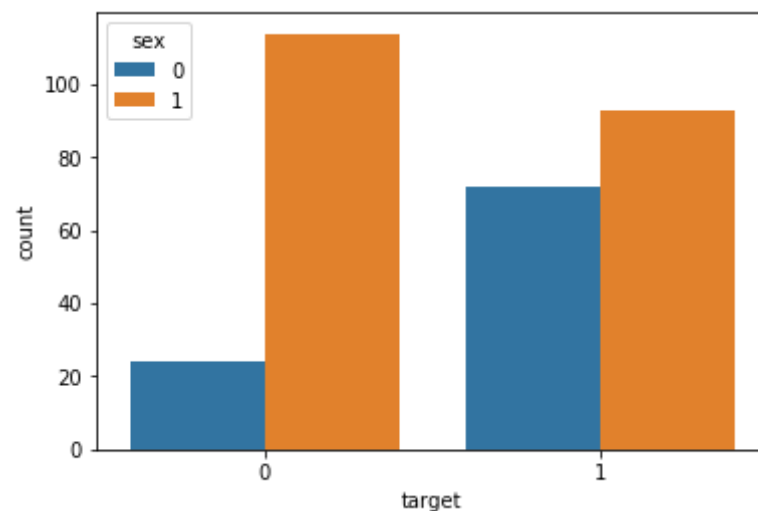
```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age          303 non-null int64
sex          303 non-null int64
cp           303 non-null int64
trestbps     303 non-null int64
chol         303 non-null int64
fbs          303 non-null int64
restecg      303 non-null int64
thalach      303 non-null int64
```

```
exang      303 non-null int64
oldpeak    303 non-null float64
slope      303 non-null int64
ca         303 non-null int64
thal       303 non-null int64
target     303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

```
In [8]: sns.countplot(x='target',hue='sex',data=df)
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x283a7a95860>
```



```
In [9]: df.describe()
```

```
Out[9]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	t
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.623762
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.937500

	age	sex	cp	trestbps	chol	fbs	restecg	t
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.5
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.0
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.0
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.0

```
In [10]: from sklearn.cross_validation import train_test_split
```

C:\Users\Dell\Anaconda3\lib\site-packages\sklearn\cross\_validation.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model\_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.  
 "This module will be removed in 0.20.", DeprecationWarning)

```
In [11]: df.columns
```

```
Out[11]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
               'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
              dtype='object')
```

```
In [12]: X=df.drop('target',axis=1)
```

```
In [13]: y=df['target']
```

```
In [14]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
In [15]: from sklearn.tree import DecisionTreeClassifier
```

```
In [16]: dtc=DecisionTreeClassifier()
```

```
In [17]: dtc.fit(X_train,y_train)
```

```
Out[17]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=N
one,
                                max_features=None, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort=False, random_state=N
one,
                                splitter='best')
```

```
In [18]: predict=dtc.predict(X_test)
```

```
In [19]: from sklearn.metrics import classification_report,confusion_matrix
```

```
In [20]: print(classification_report(y_test,predict))
print('\n')
print(confusion_matrix(y_test,predict))
```

	precision	recall	f1-score	support
0	0.68	0.57	0.62	44
1	0.65	0.74	0.69	47
avg / total	0.66	0.66	0.66	91

```
[[25 19]
 [12 35]]
```

```
In [19]: # trying with random forest
```

```
In [21]: from sklearn.ensemble import RandomForestClassifier
```

```
C:\Users\Dell\Anaconda3\lib\site-packages\sklearn\ensemble\weight_boosting.py:29: DeprecationWarning: numpy.core.umath_tests is an internal NumPy module and should not be imported. It will be removed in a future NumPy release.
from numpy.core.umath_tests import inner1d
```

```
In [22]: rfc=RandomForestClassifier(n_estimators=200) # if data is small chose estimators with low like 200
```

```
In [23]: rfc.fit(X_train,y_train)
```

```
Out[23]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=200, n_jobs=1,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)
```

```
In [24]: predictions=rfc.predict(X_test)
```

```
In [25]: print(classification_report(y_test,predictions))
print('\n')
print(confusion_matrix(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.86	0.82	0.84	44
1	0.84	0.87	0.85	47
avg / total	0.85	0.85	0.85	91

```
[[36  8]
 [ 6 41]]
```

```
In [ ]: # here random forest is fits very well
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```