



```
In [13]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
```

```
In [3]: pwd
```

```
Out[3]: 'C:\\\\Users\\admin'
```

```
In [6]: df = pd.read_csv('C:/Users/admin/Desktop/Heart.csv')
```

```
In [7]: df.head()
```

```
Out[7]:
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng
0	1	63	1	typical	145	233	1	2	150	
1	2	67	1	asymptomatic	160	286	0	2	108	
2	3	67	1	asymptomatic	120	229	0	2	129	
3	4	37	1	nonanginal	130	250	0	0	187	
4	5	41	0	nontypical	130	204	0	2	172	

```
In [8]: df.shape #303, 15
```

```
Out[8]: (303, 15)
```

```
In [9]: df.isnull().sum()
```

```
Out[9]: Unnamed: 0      0
Age      0
Sex      0
ChestPain 0
RestBP   0
Chol     0
Fbs      0
RestECG  0
MaxHR    0
ExAng    0
Oldpeak  0
Slope    0
Ca       4
Thal     2
AHD      0
dtype: int64
```

```
In [10]: df.count()
```

```
Out[10]: Unnamed: 0    303
         Age          303
         Sex          303
         ChestPain    303
         RestBP       303
         Chol         303
         Fbs          303
         RestECG      303
         MaxHR        303
         ExAng        303
         Oldpeak      303
         Slope        303
         Ca           299
         Thal         301
         AHD          303
         dtype: int64
```

```
In [11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 15 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Unnamed: 0      303 non-null   int64
1   Age             303 non-null   int64
2   Sex             303 non-null   int64
3   ChestPain       303 non-null   object
4   RestBP          303 non-null   int64
5   Chol            303 non-null   int64
6   Fbs             303 non-null   int64
7   RestECG         303 non-null   int64
8   MaxHR           303 non-null   int64
9   ExAng           303 non-null   int64
10  Oldpeak         303 non-null   float64
11  Slope           303 non-null   int64
12  Ca              299 non-null   float64
13  Thal            301 non-null   object
14  AHD             303 non-null   object
dtypes: float64(2), int64(10), object(3)
memory usage: 35.6+ KB
```

```
In [12]: df.dtypes
```

```
Out[12]: Unnamed: 0      int64
Age      int64
Sex      int64
ChestPain object
RestBP   int64
Chol     int64
Fbs      int64
RestECG  int64
MaxHR    int64
ExAng    int64
Oldpeak  float64
Slope    int64
Ca       float64
Thal     object
AHD      object
dtype: object
```

```
In [14]: df==0
```

```
Out[14]:
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	E
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	True	False	False	
2	False	False	False	False	False	False	True	False	False	
3	False	False	False	False	False	False	True	True	False	
4	False	False	True	False	False	False	True	False	False	
...
298	False	False	False	False	False	False	True	True	False	
299	False	False	False	False	False	False	False	True	False	
300	False	False	False	False	False	False	True	True	False	
301	False	False	True	False	False	False	True	False	False	
302	False	False	False	False	False	False	True	True	False	

303 rows × 15 columns

```
In [15]: df[df==0]
```

```
Out[15]:
```

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	NaN
4	NaN	NaN	0.0	NaN	NaN	NaN	0.0	NaN	NaN	NaN
...
298	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	NaN
299	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN
300	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	NaN
301	NaN	NaN	0.0	NaN	NaN	NaN	0.0	NaN	NaN	NaN
302	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.0	NaN	NaN

303 rows × 15 columns

```
In [16]: (df == 0).sum()
```

```
Out[16]: Unnamed: 0      0
Age      0
Sex      97
ChestPain 0
RestBP   0
Chol     0
Fbs     258
RestECG  151
MaxHR    0
ExAng    204
Oldpeak  99
Slope    0
Ca      176
Thal     0
AHD     0
dtype: int64
```

```
In [17]: np.mean(df['Age'])
```

```
Out[17]: 54.43894389438944
```

```
In [18]: df.Age.mean()
```

```
Out[18]: 54.43894389438944
```

```
In [19]: df.columns
```

```
Out[19]: Index(['Unnamed: 0', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs',  
              'RestECG', 'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal', 'AHD'],  
              dtype='object')
```

```
In [20]: data = df[['Age', 'Sex', 'ChestPain', 'RestBP', 'Chol']]
```

```
In [21]: train,test = train_test_split(data,test_size=0.25,random_state=1)
```

```
In [22]: train.shape
```

```
Out[22]: (227, 5)
```

```
In [23]: test.shape
```

```
Out[23]: (76, 5)
```

```
In [24]: actual = np.concatenate((np.ones(45),np.zeros(450),np.ones(5)))  
actual
```

```
Out[24]: array([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., 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., 0.,  
                0., 0., 1., 1., 1., 1., 1.]
```

```
In [25]: predicted = np.concatenate((np.ones(100),np.zeros(400)))  
predicted
```

[illegible]

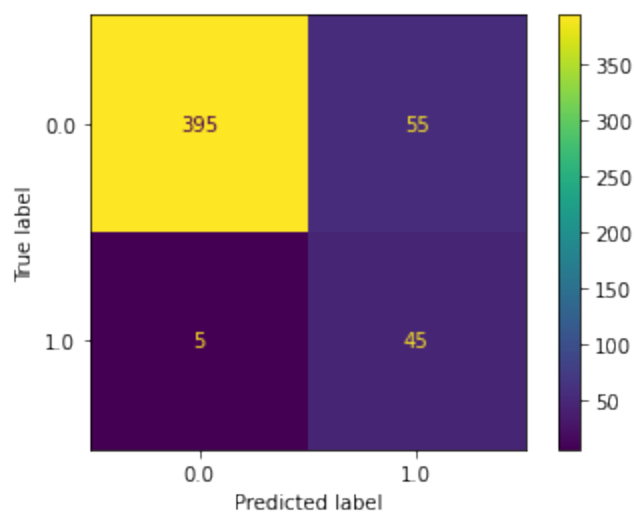
```
In [26]: type(predicted)
```

Out[26]: numpy.ndarray

```
In [27]: from sklearn.metrics import ConfusionMatrixDisplay
```

```
In [28]: ConfusionMatrixDisplay.from_predictions(actual, predicted)
```

```
Out[28]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1b8057c6b
b0>
```



```
In [29]: from sklearn.metrics import classification_report
         from sklearn.metrics import accuracy_score
```

```
In [30]: print(classification_report(actual, predicted))
```

	precision	recall	f1-score	support
0.0	0.99	0.88	0.93	450
1.0	0.45	0.90	0.60	50
accuracy			0.88	500
macro avg	0.72	0.89	0.76	500
weighted avg	0.93	0.88	0.90	500

```
In [ ]: accuracy_score(actual, predicted)
```