**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

df **=** pd**.**read\_csv('../input/heartcsv/Heart.csv')

df**.**head()

df**.**shape *#303, 15*

df**.**isnull()**.**sum()

df**.**count()

df**.**info()

df**==**0

:

df[df**==**0]

(df **==** 0)**.**sum()

dtype: int64

np**.**mean(df['Age'])

df**.**Age**.**mean()

df**.**columns

data **=** df[['Age', 'Sex', 'ChestPain', 'RestBP', 'Chol']]

*#Cross validation*

train,test **=** train\_test\_split(data,test\_size**=**0.25,random\_state**=**1)

train**.**shape

test**.**shape

(76, 5)

actual **=** np**.**concatenate((np**.**ones(45),np**.**zeros(450),np**.**ones(5)))

*# run = np.array([1,0,1,1,1])*

predicted **=** np**.**concatenate((np**.**ones(100),np**.**zeros(400)))

predicted

numpy.ndarray

**from** sklearn.metrics **import** ConfusionMatrixDisplay

ConfusionMatrixDisplay**.**from\_predictions(actual,predicted)

**from** sklearn.metrics **import** classification\_report

**from** sklearn.metrics **import** accuracy\_score

print(classification\_report(actual,predicted))

accuracy\_score(actual,predicted)