1. Can you get the feature names and the descriptions of the toy diabetes data set via python code? Please write down the code and the output of the code. [1 point]

Code:

from sklearn.datasets import load\_diabetes

data\_bunch = load\_diabetes()

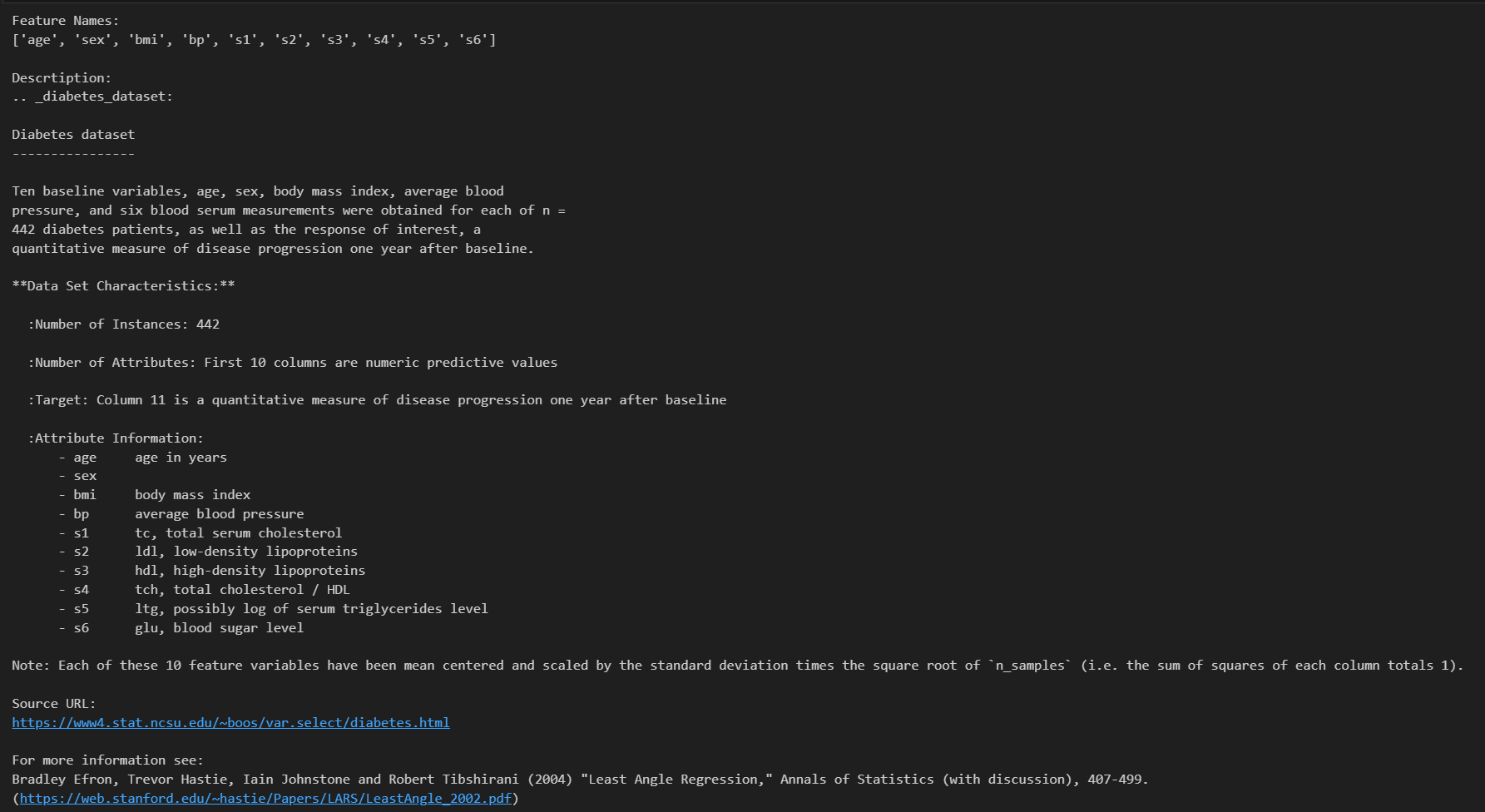
print("Feature Names:")

print(data\_bunch.feature\_names)

print("\nDescrtiption:")

print(data\_bunch.DESCR)

Output:



2. It is time to play with another data set of scikit\_learn: the wine data set. Use **load\_wine**() to load the wine data set and apply logistic regression on it. We split the data set into 80/20. 80% of data points are used for model training, and 20% for test purposes. Please write the code and output (including mean accuracy and confusion matrix) in the answer. [2 points]

Code:

from sklearn.datasets import load\_wine

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import plot\_confusion\_matrix

import matplotlib.pyplot as plt

import numpy as np

def warn(\*args, \*\*kwargs):

pass

import warnings

warnings.warn = warn

data\_bunch = load\_wine()

print("Feature Names")

print(data\_bunch.feature\_names)

print("\nClasses")

print(data\_bunch.target\_names)

wine\_X, wine\_y = data\_bunch.data, data\_bunch.target

def filter\_class\_2(X, y):

new\_X, new\_y = [], []

for x, y in zip(X, y):

if y == 2:

continue

else:

new\_X.append(x)

new\_y.append(y)

return np.array(new\_X), np.array(new\_y)

wine\_X, wine\_y = filter\_class\_2(wine\_X, wine\_y)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(wine\_X, wine\_y, test\_size=0.2)

logit\_regr = LogisticRegression()

logit\_regr.fit(x\_train, y\_train)

logit\_regr.classes\_

print("\nAccuracy Score")

print(logit\_regr.score(x\_test, y\_test))

plot\_confusion\_matrix(logit\_regr, x\_test, y\_test)

plt.show()

Output:

