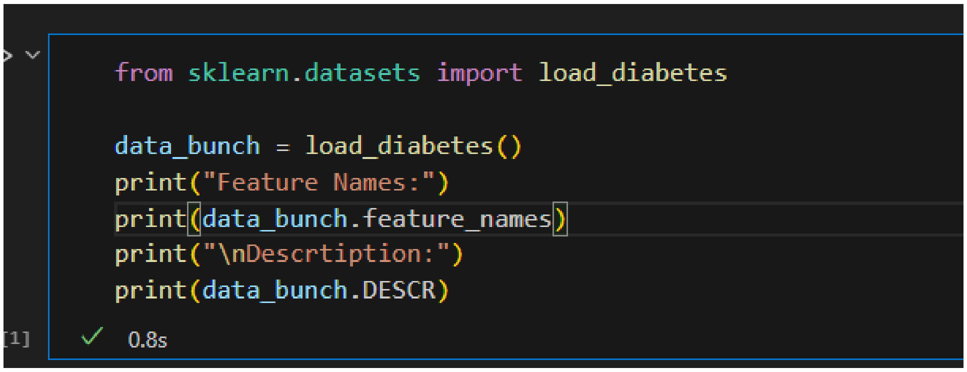
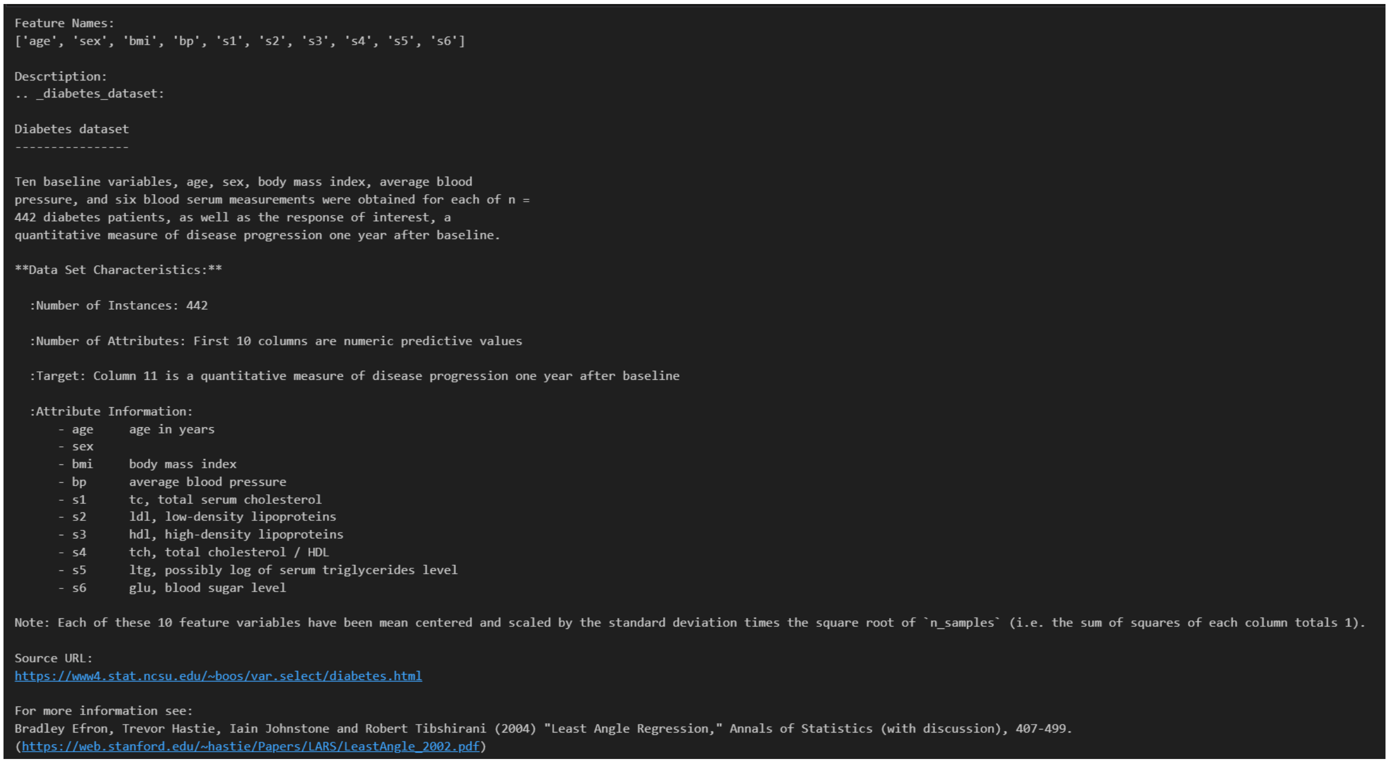
1. Can you get the feature names and the descriptions of the toy diabetes data set via python code? Please write down the code and the output of the code. [1 point]

Code:



```python
from sklearn.datasets import load_diabetes

data_bunch = load_diabetes()
print("Feature Names:")
print(data_bunch.feature_names)
print("\nDescrtiption:")
print(data_bunch.DESCR)
```

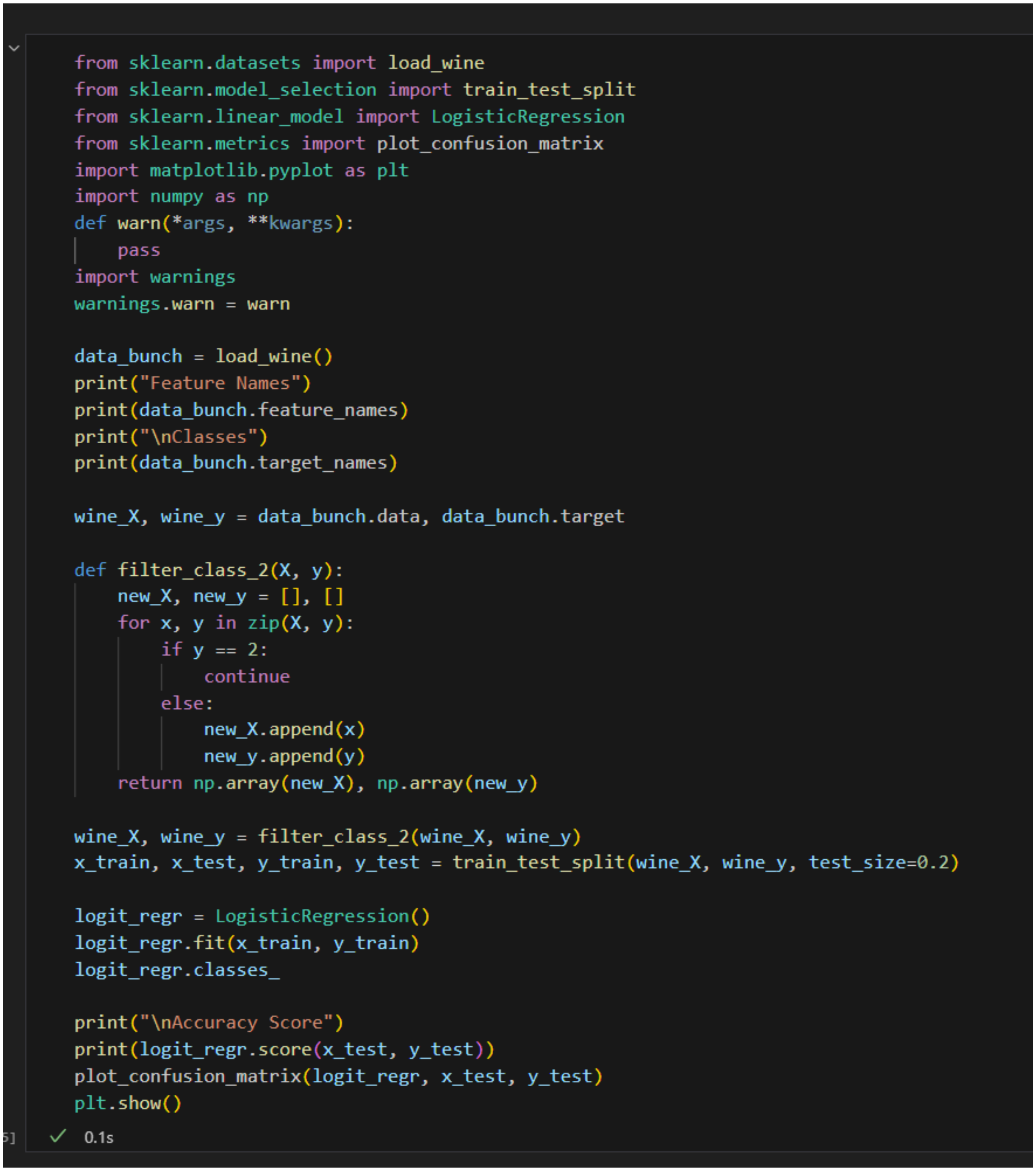
Output:



```
Feature Names:
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']

Descrtiption:
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values

  :Target: Column 11 is a quantitative measure of disease progression one year after baseline

  :Attribute Information:
      - age     age in years
      - sex
      - bmi     body mass index
      - bp      average blood pressure
      - s1      tc, total serum cholesterol
      - s2      ldl, low-density lipoproteins
      - s3      hdl, high-density lipoproteins
      - s4      tch, total cholesterol / HDL
      - s5      ltg, possibly log of serum triglycerides level
      - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times the square root of `n_samples` (i.e. the sum of squares of each column totals 1).

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of Statistics (with discussion), 407-499.
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```

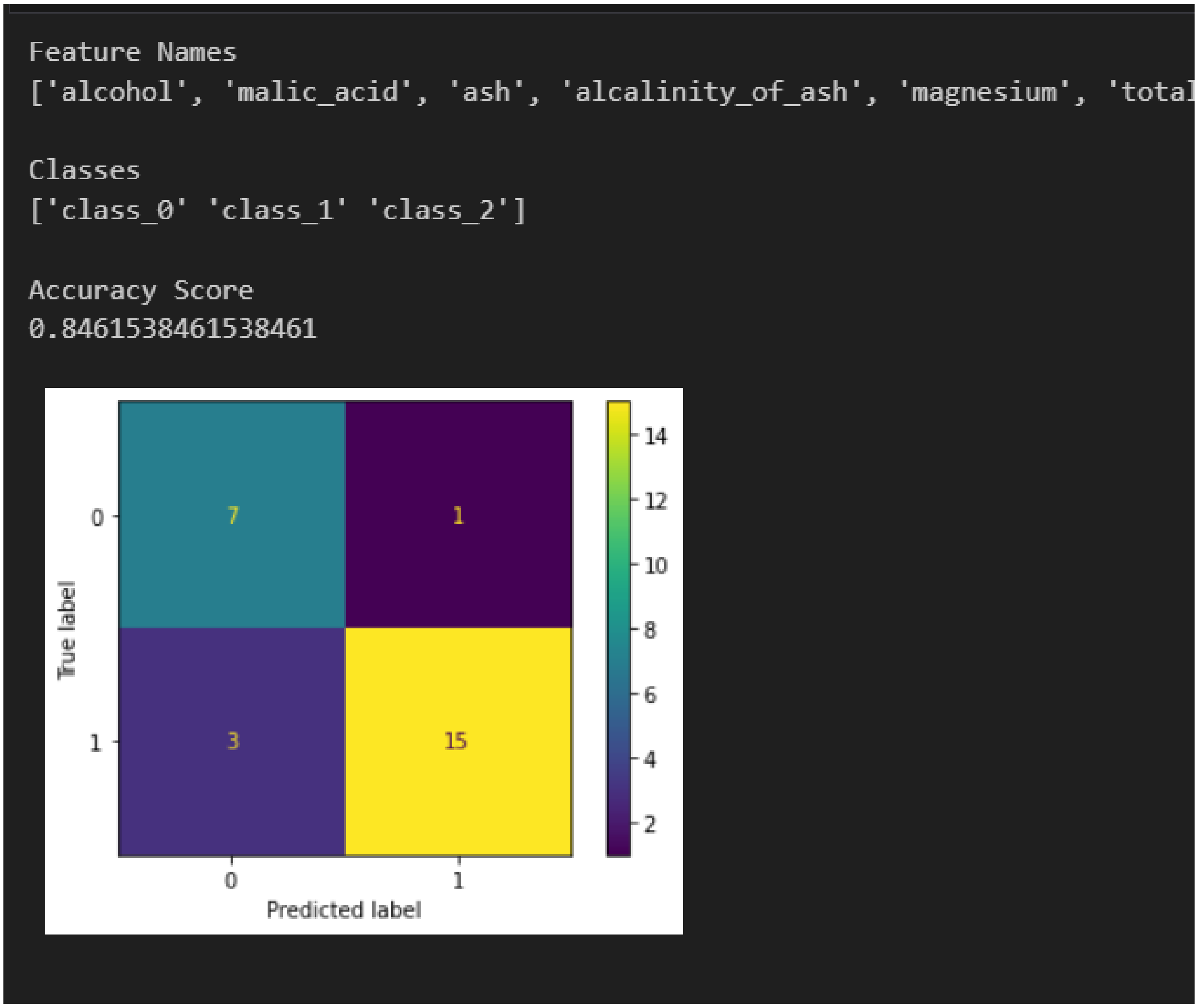
2. It is time to play with another data set of scikit_learn: the wine data set. Use **load_wine**() to load the wine data set and apply logistic regression on it. We split the data set into 80/20. 80% of data points are used for model training, and 20% for test purposes. Please write the code and output (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:



```
Feature Names
['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total

Classes
['class_0' 'class_1' 'class_2']

Accuracy Score
0.8461538461538461
```