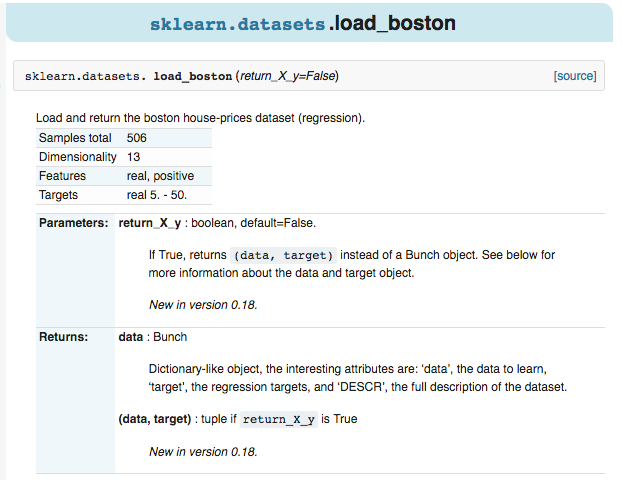
0以鸢尾花的特征作为数据来源，数据集包含150个数据集，分为3类，每类50个数据，每个数据包含4个属性，是在数据挖掘、数据分类中非常常用的测试集、训练集

三类分别为:setosa, versicolor, virginica

数据包含4个独立的属性,这些属性变量测量植物的花朵,比如萼片长度, 萼片宽度,花瓣长度, 花瓣宽度。



导入 datasets 包，本文以 Linear Regression 为例。

from \_\_future\_\_ import print\_function

from sklearn import datasets

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

#### 导入数据－训练模型

用 datasets.load\_boston() 的形式加载数据，并给 X 和 y 赋值，这种形式在 Sklearn 中都是高度统一的。

loaded\_data = datasets.load\_boston()

data\_X = loaded\_data.data

data\_y = loaded\_data.target

定义模型。

可以直接用默认值去建立 model，默认值也不错，也可以自己改变参数使模型更好。   
然后用 training data 去训练模型。

model = LinearRegression()

model.fit(data\_X, data\_y)

再打印出预测值，这里用 X 的前 4 个来预测，同时打印真实值，作为对比，可以看到是有些误差的。

print(model.predict(data\_X[:4, :]))

print(data\_y[:4])

[ 30.00821269 25.0298606 30.5702317 28.60814055]

[ 24. 21.6 34.7 33.4]

为了提高准确度，可以通过尝试不同的 model，不同的参数，不同的预处理等方法，入门的话可以直接用默认值。

#### 创建虚拟数据－可视化

下面是创造数据的例子。

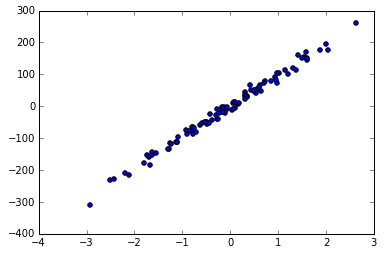
用函数来建立 100 个 sample，有一个 feature，和一个 target，这样比较方便可视化。

X, y = datasets.make\_regression(n\_samples=100, n\_features=1, n\_targets=1, noise=10)

用 scatter 的形式来输出结果。

plt.scatter(X, y)

plt.show()



可以看到用函数生成的 Linear Regression 用的数据。

noise 越大的话，点就会越来越离散，例如 noise 由 10 变为 50.

X, y = datasets.make\_regression(n\_samples=100, n\_features=1, n\_targets=1, noise=50)

plt.scatter(X, y)

plt.show()

