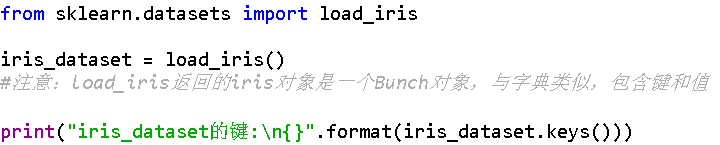
1. 背景：

之前我们用简单的二分类方法，实现了对鸢尾花品种setosa、virginica的分类，现在我们来扩展一下，实现对setosa、virginica和versicolor的三分类。

1. 数据集

鸢尾花数据集是一个经典的数据集，它包含在scikit-learn的datasets模块中，可以调用load\_iris函数来加载数据：

利用load\_iris()方法即可加载鸢尾花数据集，只是返回的是Bunch对象，利用字典的方法，查看Bunch对象中的键。





可以看到，Bunch对象的键有：

data：n行m列的ndarray数组，每一行代表一朵花，每一列代表该花的一个特征。

target：每朵花的品种，一维数组

target\_names：类名列表，ndarray类型

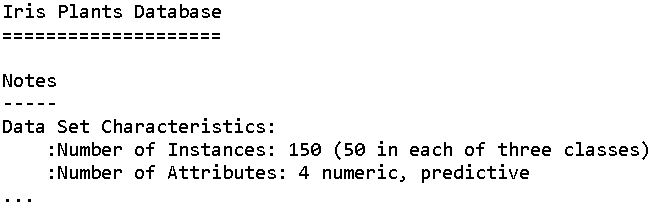
DESCR：数据集的简要说明

feature\_names：字符串列表，对每一个特征进行了说明。

下面，我们分别查看每一个键的值：

1. DESCR





可以看到有相关数据集的一些说明，比如上面的数据集有150个样本，每个样本的属性值为4。

1. target\_names

target\_names对象的值是一个字符串列表，里面包含了待预测的花的品种，是一个ndarray类型的数组。





1. feature\_names

feature\_names键对应的值是一个字符串列表，对每一个特征进行了说明：





可以看到一共有4个特征，分别是：花萼长度、花萼宽度、花瓣长度和花瓣宽度。

1. data与target

数据包含在target和data字段中，data里面是是NumPy数组格式的花萼长度、花萼宽度、花瓣长度和花瓣宽度。Data数据的每一行代表一朵花，列代表花的四个测量数据：

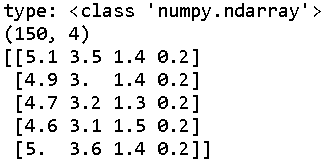




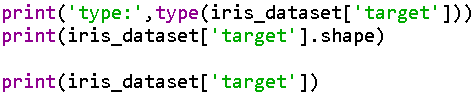
可以看到data数组有150行4列。

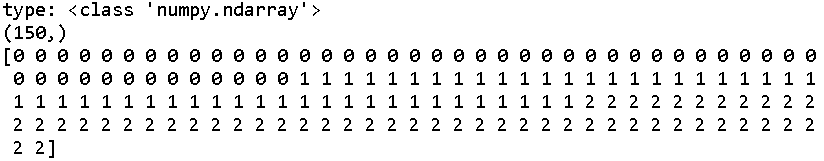
可以看下前5行的值：





target是一维数组，每朵花对应其中的一个数据：





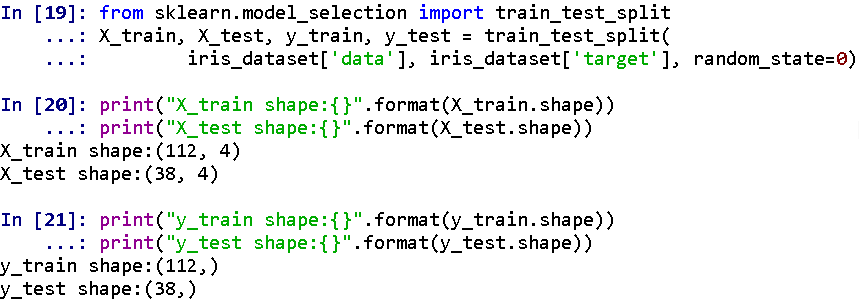
可以看到target中按照数据的标签排列，其中0代表山鸢尾setosa，1代表变色鸢尾versicolor，2代表virginica。

1. 训练数据与测试数据

scikit-learn中的train\_test\_split函数可以打乱数据集并进行拆分。将75%的行数据及对应的标签作为训练集，剩下的25%的数据及标签作为测试集。

scikit-learn中的数据通常是大写字母X表示，而标签用小写的y表示。X是一个（二维数组）矩阵，y是一个一维数组（向量）。

train\_test\_split函数利用伪随机数生成器将数据集打乱，random\_state参数指定了随机数生成器的种子，确保多次运行同一函数能够得到相同的输出。

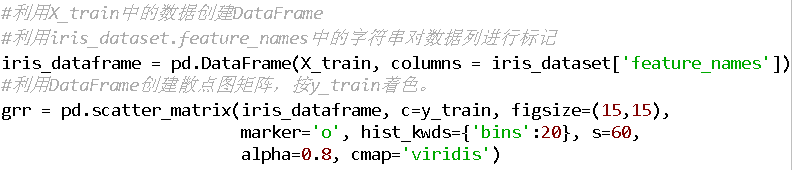


可以看到用于训练集的数据有112个，剩下的38个用于测试集。

1. 构建模型前的准备：观察数据
   1. 可以查看所需信息是否包含在数据中
   2. 可以发现异常值和特殊值
   3. 对于二特征数据，可以使用二维散点图，将一个特征作为x轴，另一个特征作为y轴，将每个数据绘制为图上的一个点。

对于多特征数据，可以使用散点图矩阵，两两查看所有特征，但是无法同时显示所有特征之间的关系。

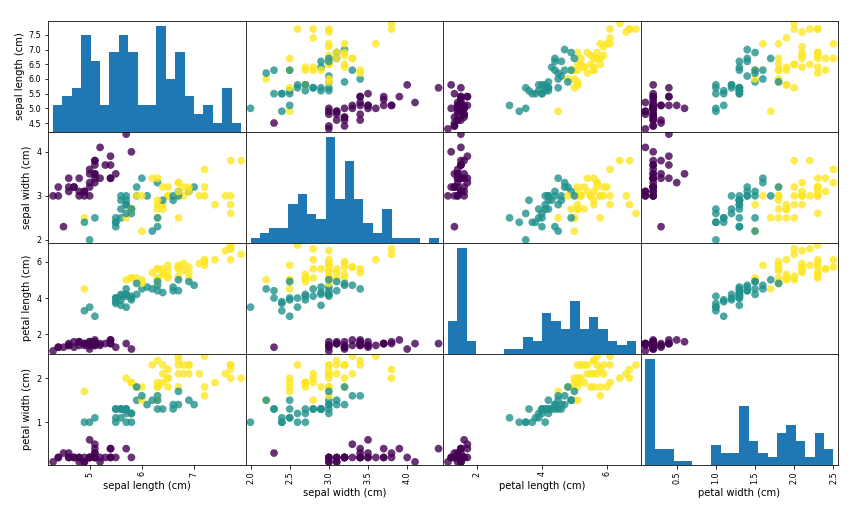
* 1. 将Numpy数组转为pandas的DataFrame对象，pandas的方法scatter\_matrix()方法用于绘制散点图矩阵，矩阵的对角线是每个特征的直方图。



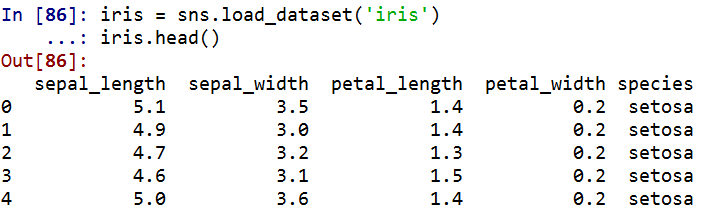
pd.scatter\_matrix(data, c=散点的颜色, figsize=图像大小, marker=’点标记’,

hist\_kwds={‘bins’:柱状图间距}, s=散点大小, alpha=透明度，

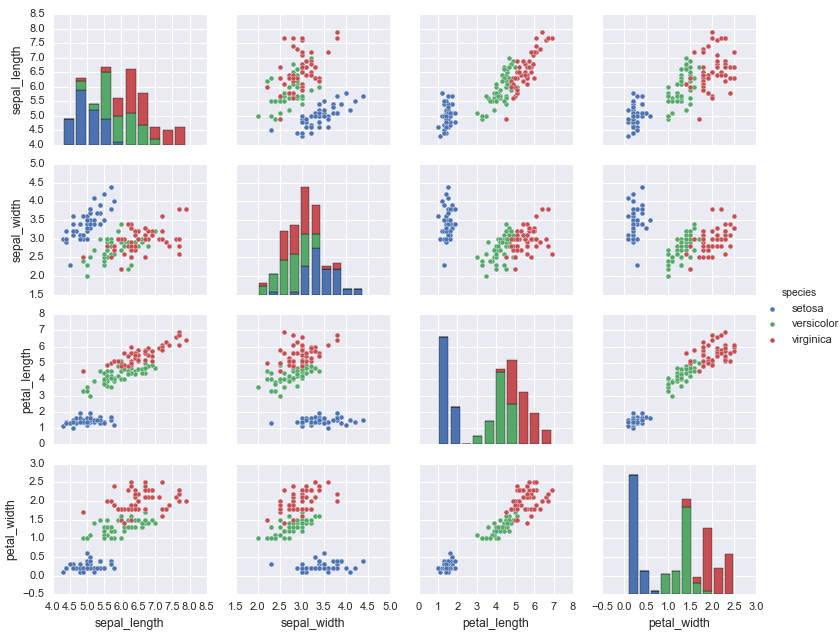
cmap颜色图)



或者利用seaborn模块中的pairplot方法：







从中可以看出，利用花萼和花瓣的测量数据基本可以将三类分开。

1. 构建第一个模型：k近邻算法
   1. 此模型只需要保存训练集即可。要对一个新的数据点作出预测，算法会在训练集中寻找与这个新数据点距离最近的数据点，然后找到的数据点的标签赋值给这个新数据点。

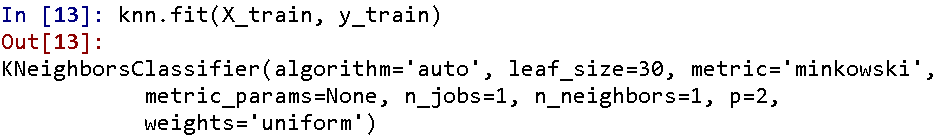
k的含义是，考虑训练集中与新数据点最近的k个邻居，而不是只考虑最近的哪一个，可以用这些邻居中数量最多的类别作出预测。

* 1. k近邻算法在neighbors模块的KNeighbors Classifier类中实现。需要设置模型参数，即邻居的数目n\_neighbors，这里设为1：



knn对象中既包含用训练数据构建模型的算法，也包含对新数据点进行预测的算法。

* 1. 利用knn.fit(X\_train, y\_train)基于训练集构建模型。

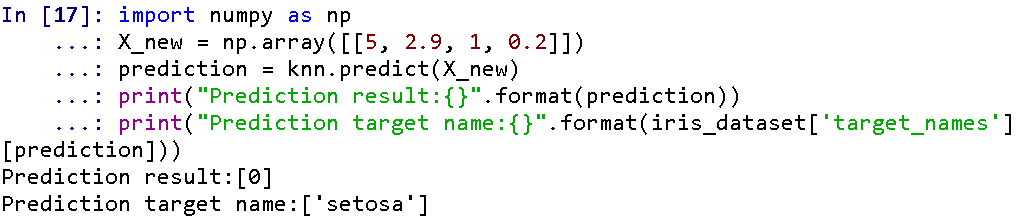


fit方法返回的是knn对象本身，从中也可以看到相关的参数设置。

六、 作出预测

1. 假设一个新的样本X\_new，其相对应的四个特征值是【5,2.9,1,0.2】我们需要将其放入数组中，注意，一定要在ndarray中给数据列表再加一个列表框，因为不加的话shape为（4，），不是二维的数组，要手动加一个框，此时的shape为（1,4），因为scikit-learn的输入数组必须是二维的。

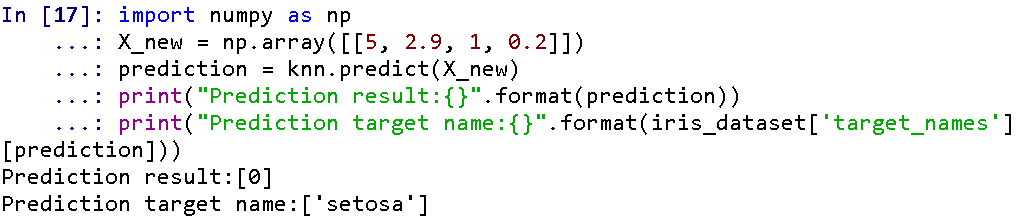
2. 然后调用knn的predict方法来进行预测：



Prediction的结果是0，表示是山鸢尾。

1. 作出假设
2. 假设一个新的样本X\_new，其相对应的四个特征值是【5,2.9,1,0.2】我们需要将其放入数组中，注意，一定要在ndarray中给数据列表再加一个列表框，因为不加的话shape为（4，），不是二维的数组，要手动加一个框，此时的shape为（1,4），因为scikit-learn的输入数组必须是二维的。

2. 然后调用knn的predict方法来进行预测：



Prediction的结果是0，表示是山鸢尾。

七、 评估模型

我们可以对测试数据中的每朵鸢尾花进行预测，并将预测结果与标签（已知的品种）进行对比。我们可以通过计算精度来衡量模型的优劣，精度就是品种预测正确的花所占的比例：

