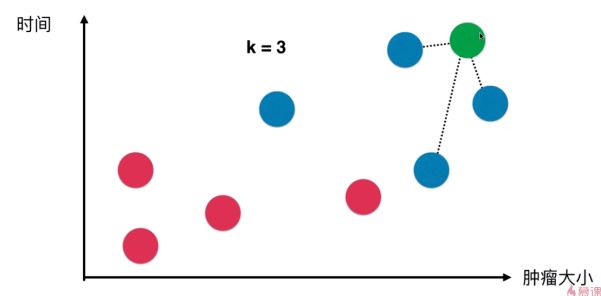
# KNN

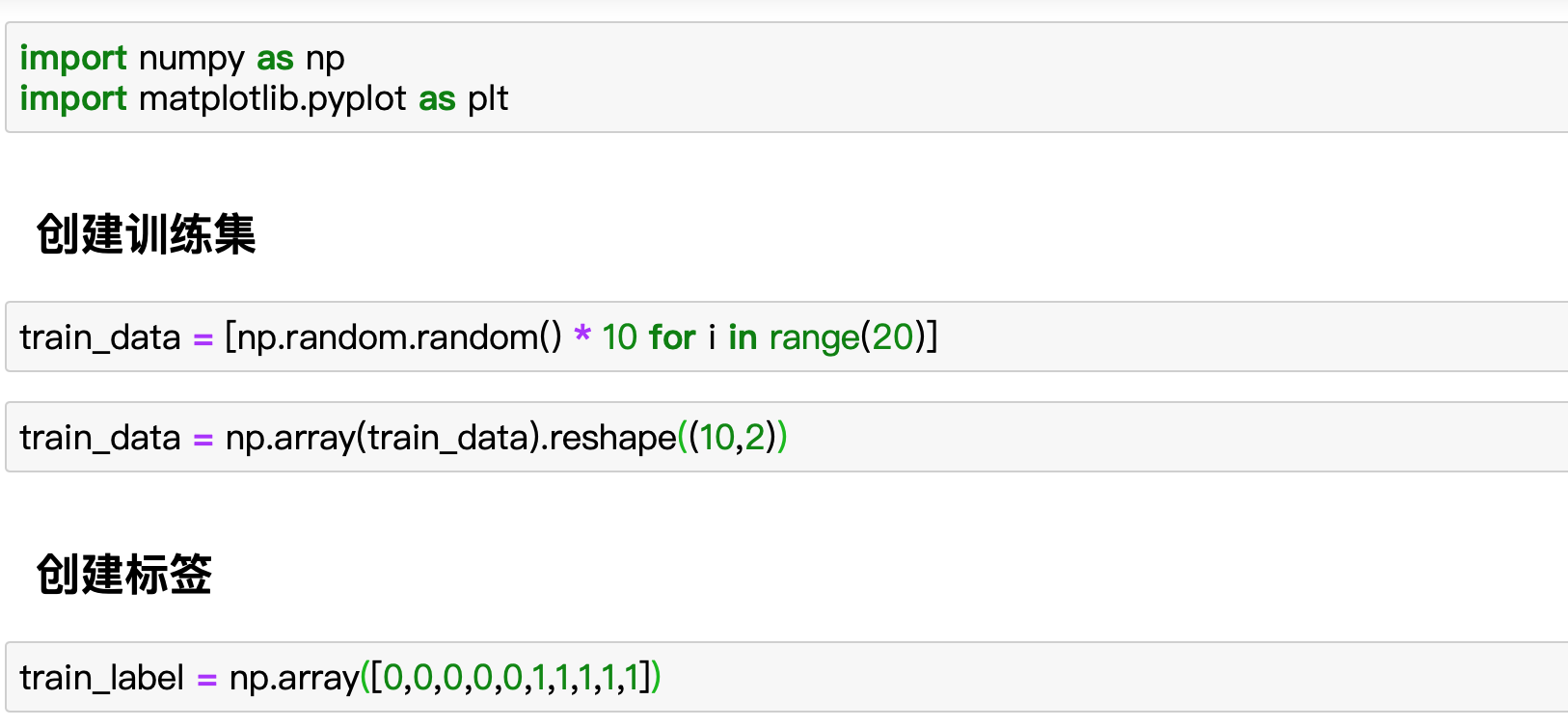
* 概述

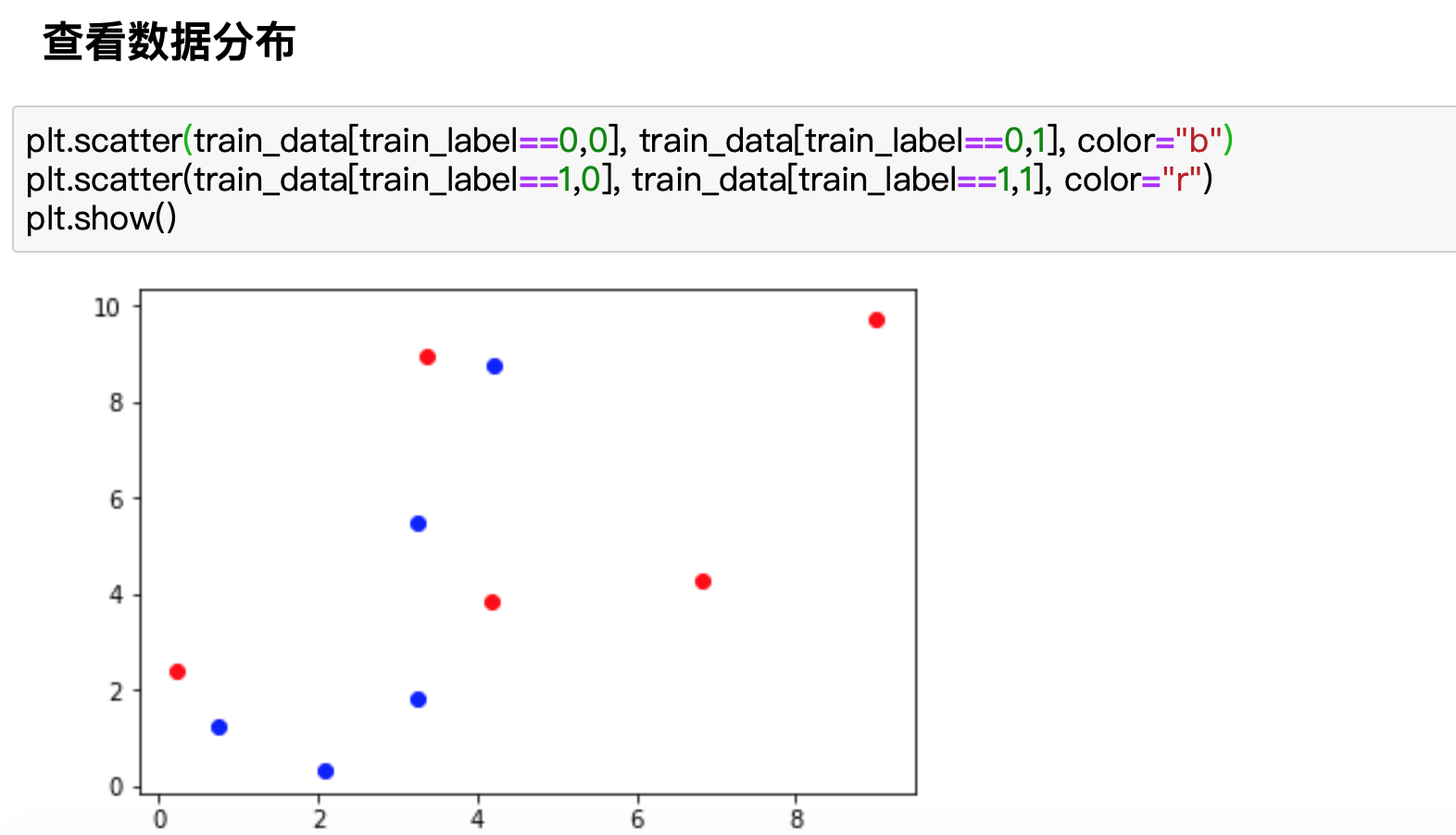
KNN(K-Nearest Neighbors),又称K-近邻算法。

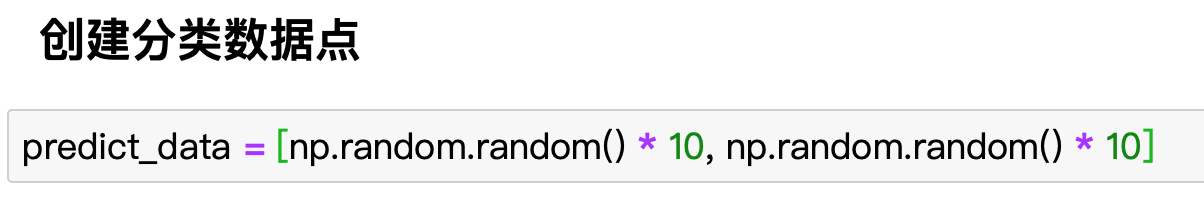
简单来讲，K-近邻算法采用不同特征值之间的距离方法进行分类。

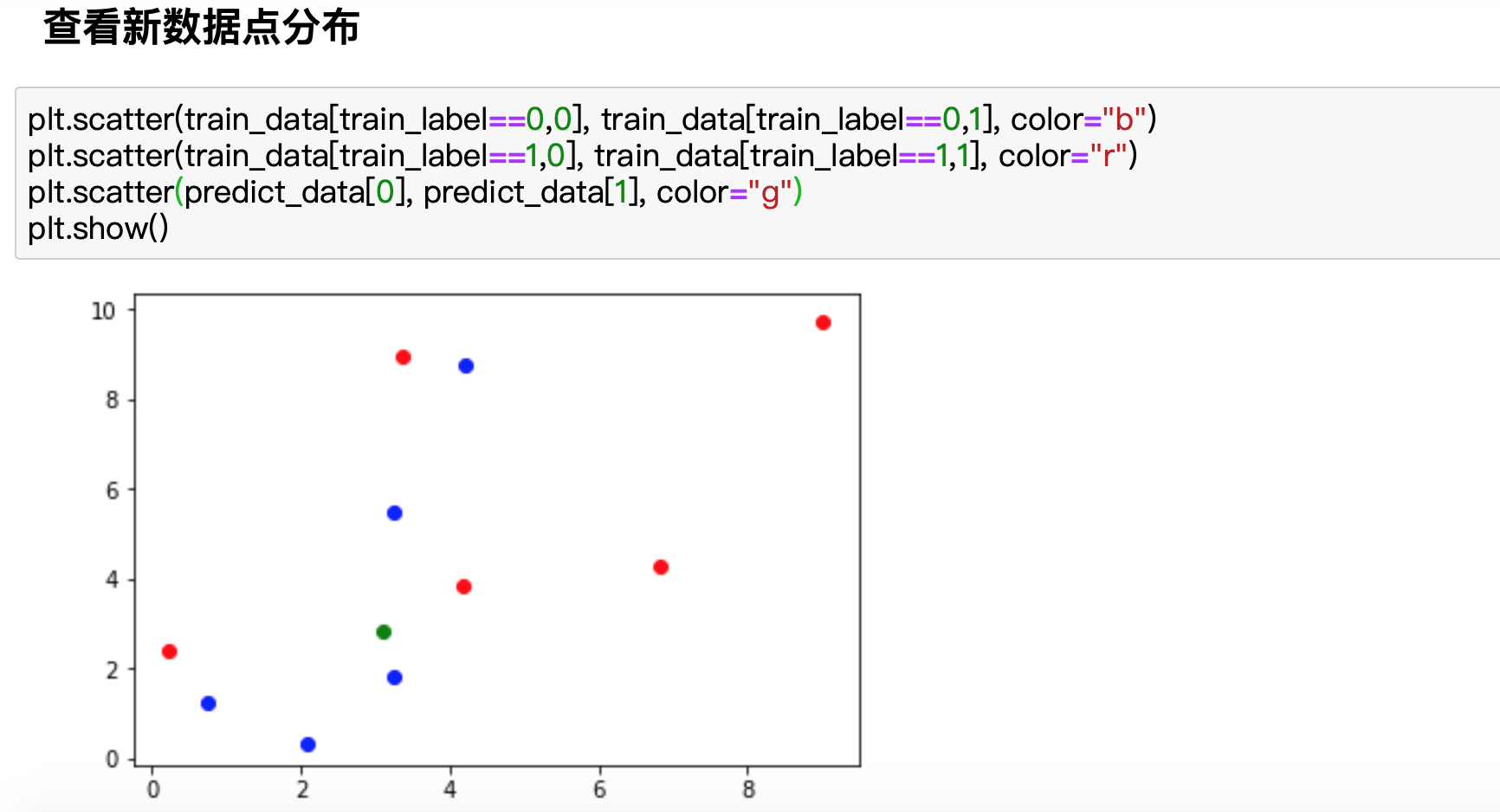


* 算法实践

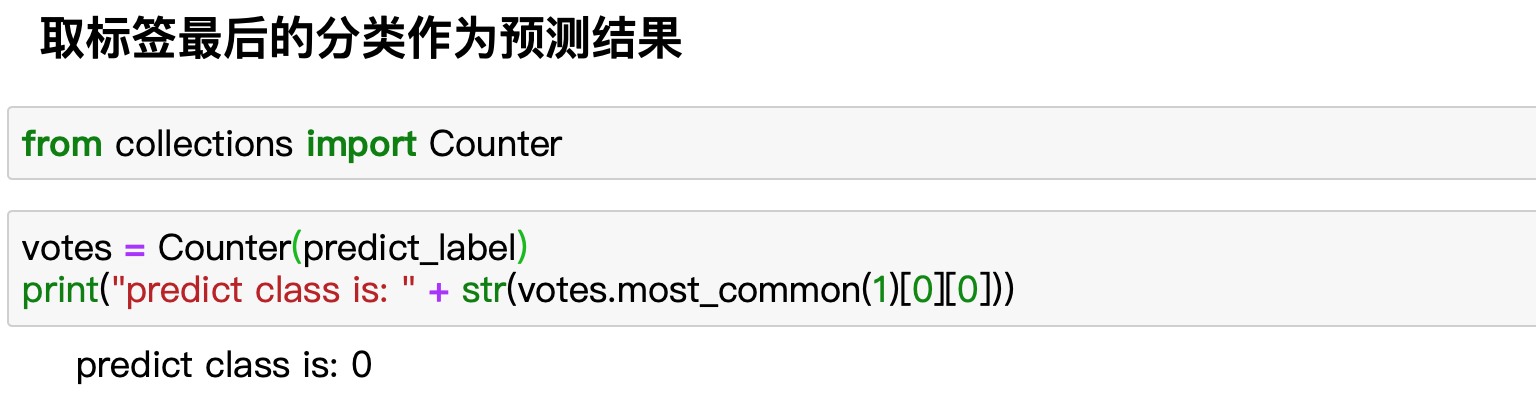




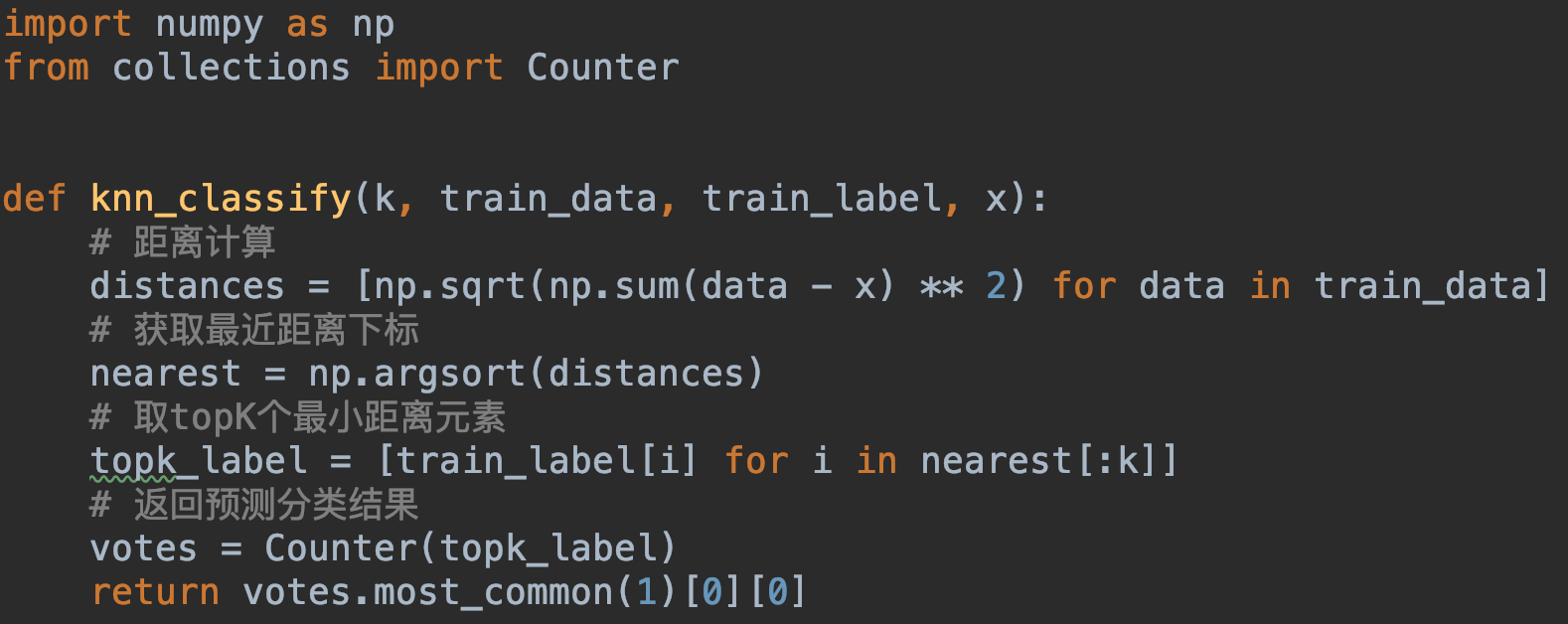






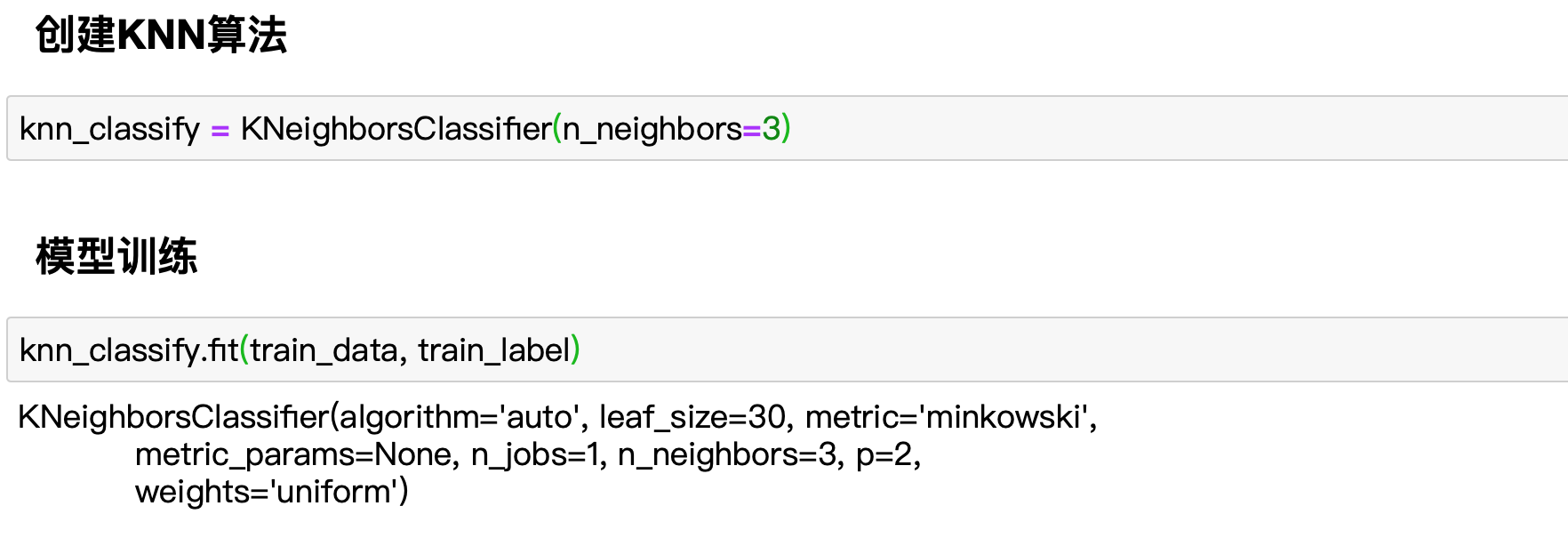


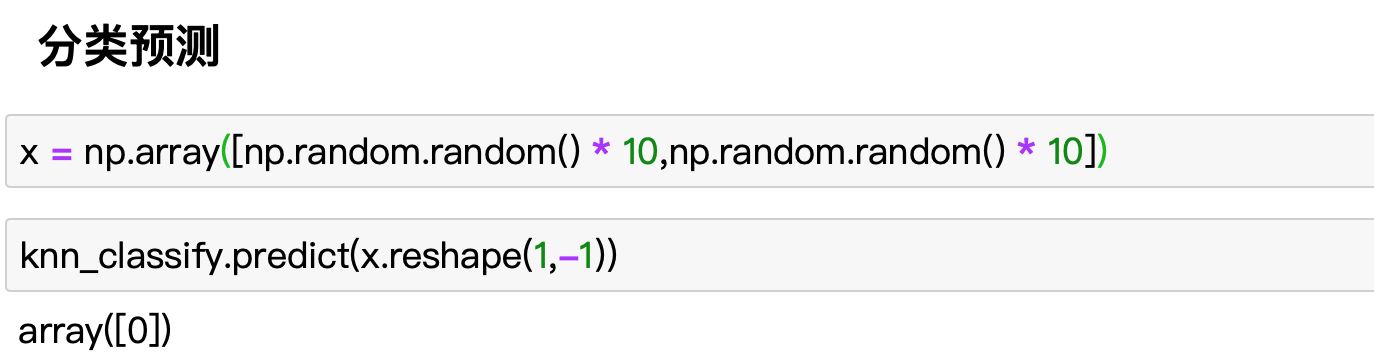
* 算法封装



* scikit-learn中的KNN





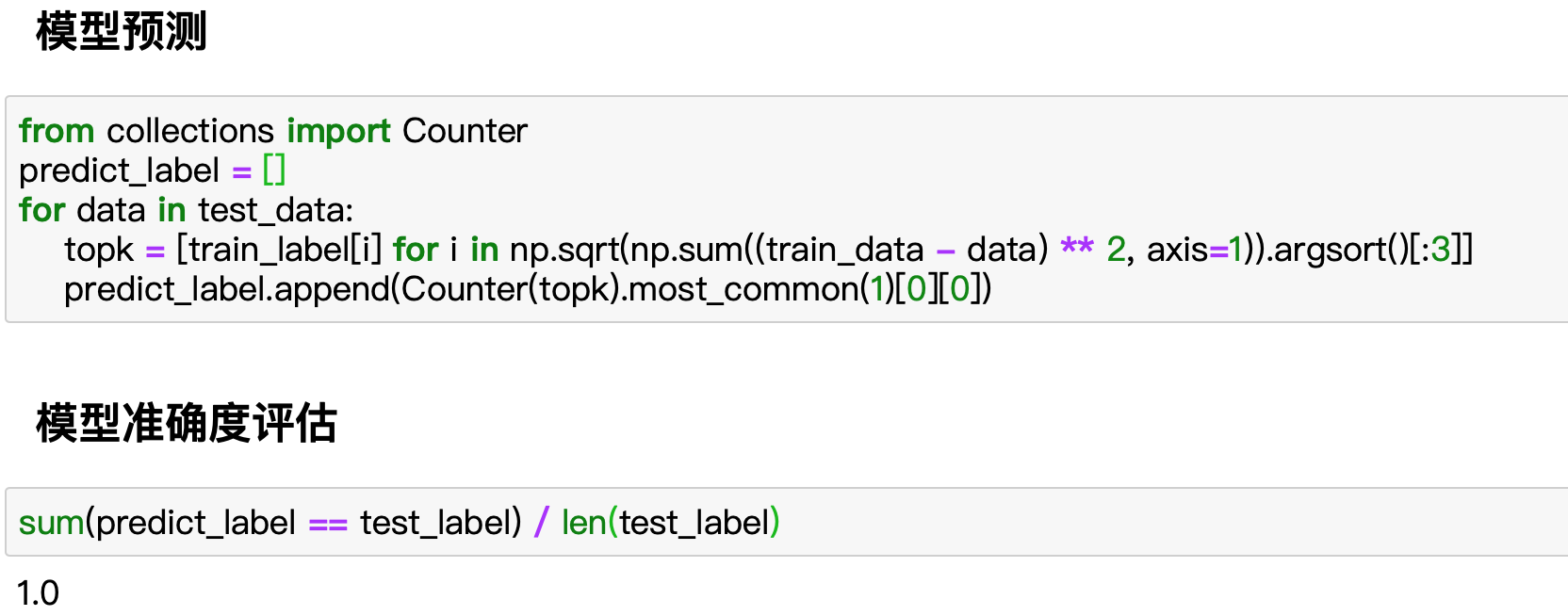


* 模型性能评估

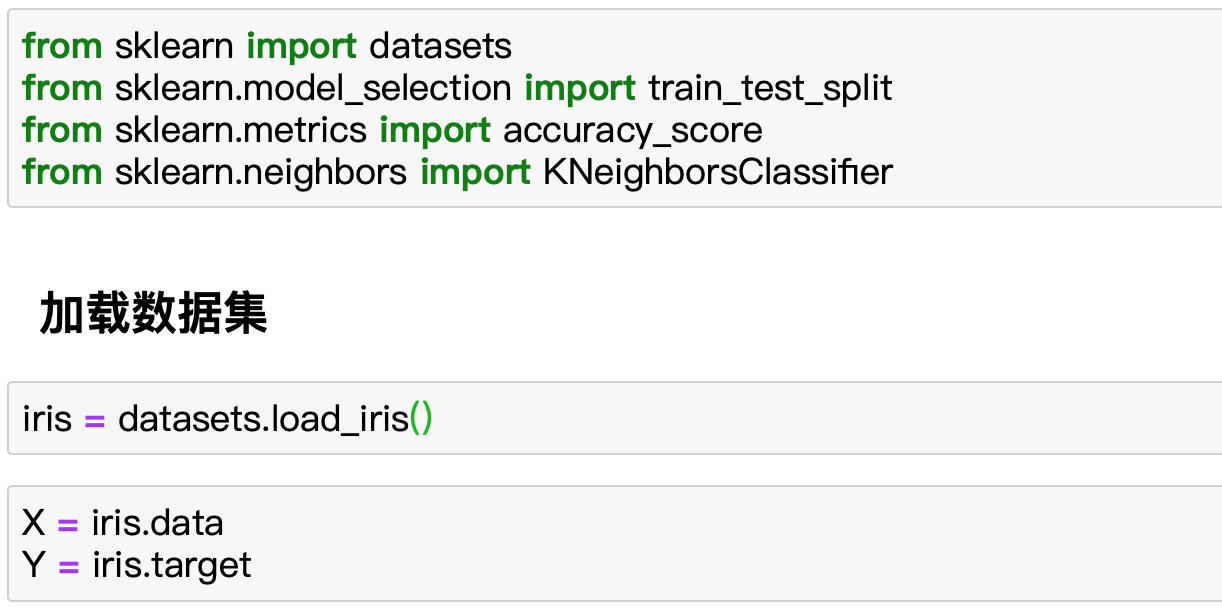
**1.train test split**

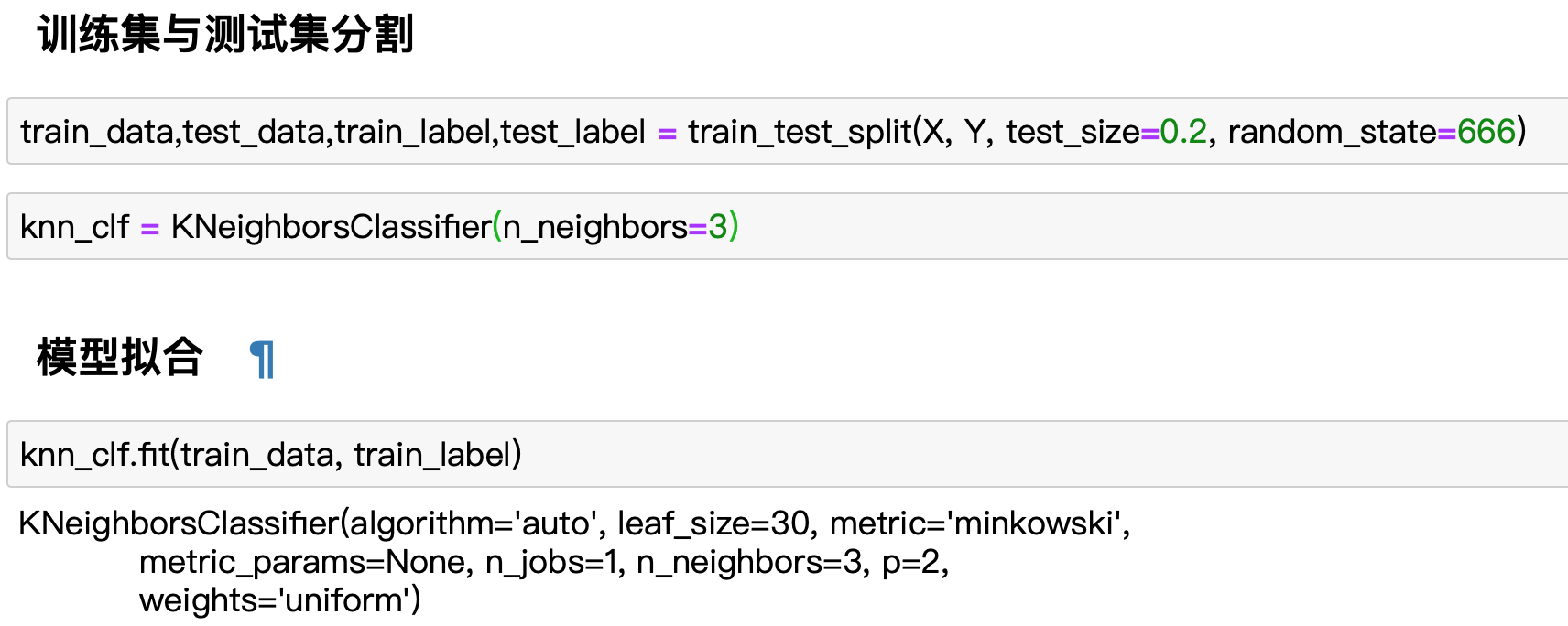
****

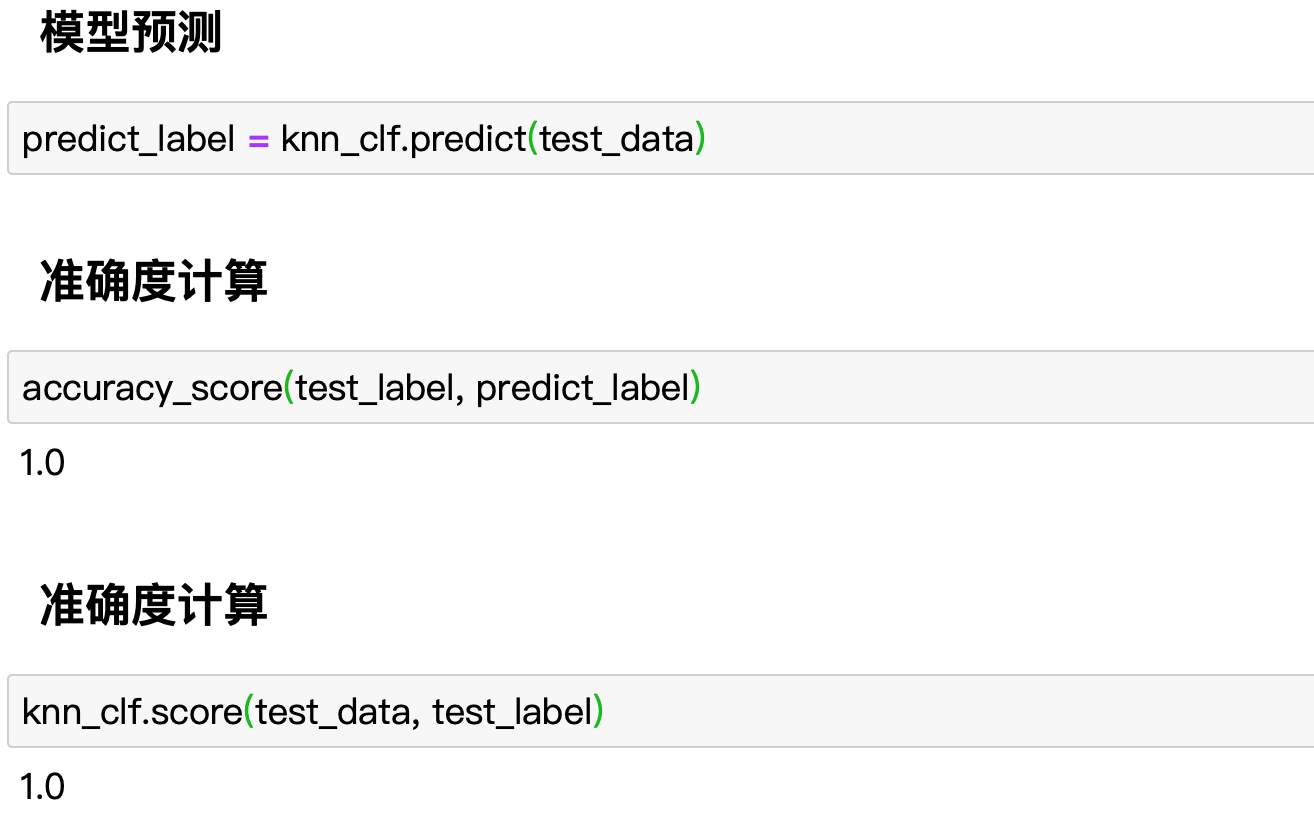
****

****

* scikit-learn中的模型性能评估







* 超参数与模型参数

**超参数：在模型运行前需要决定的参数**

**模型参数：算法过程中学习的参数**

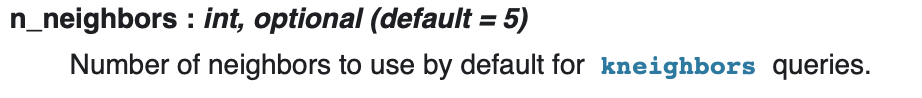
**KNN算法中没有模型参数**

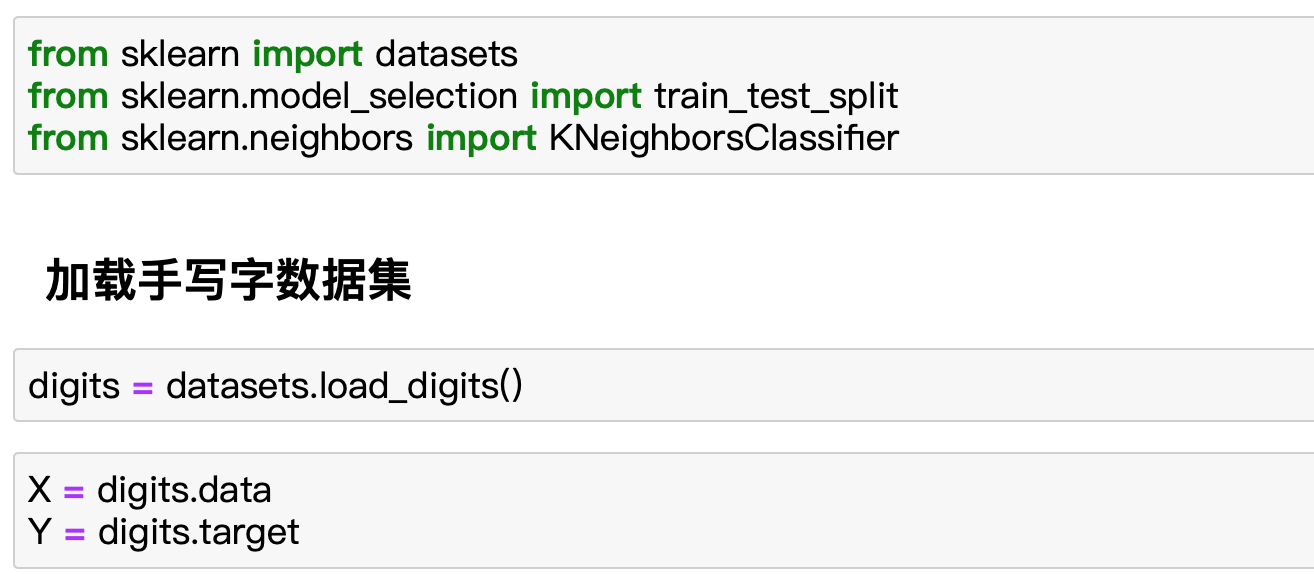
**KNN算法中n\_neighbors是典型的超参数**

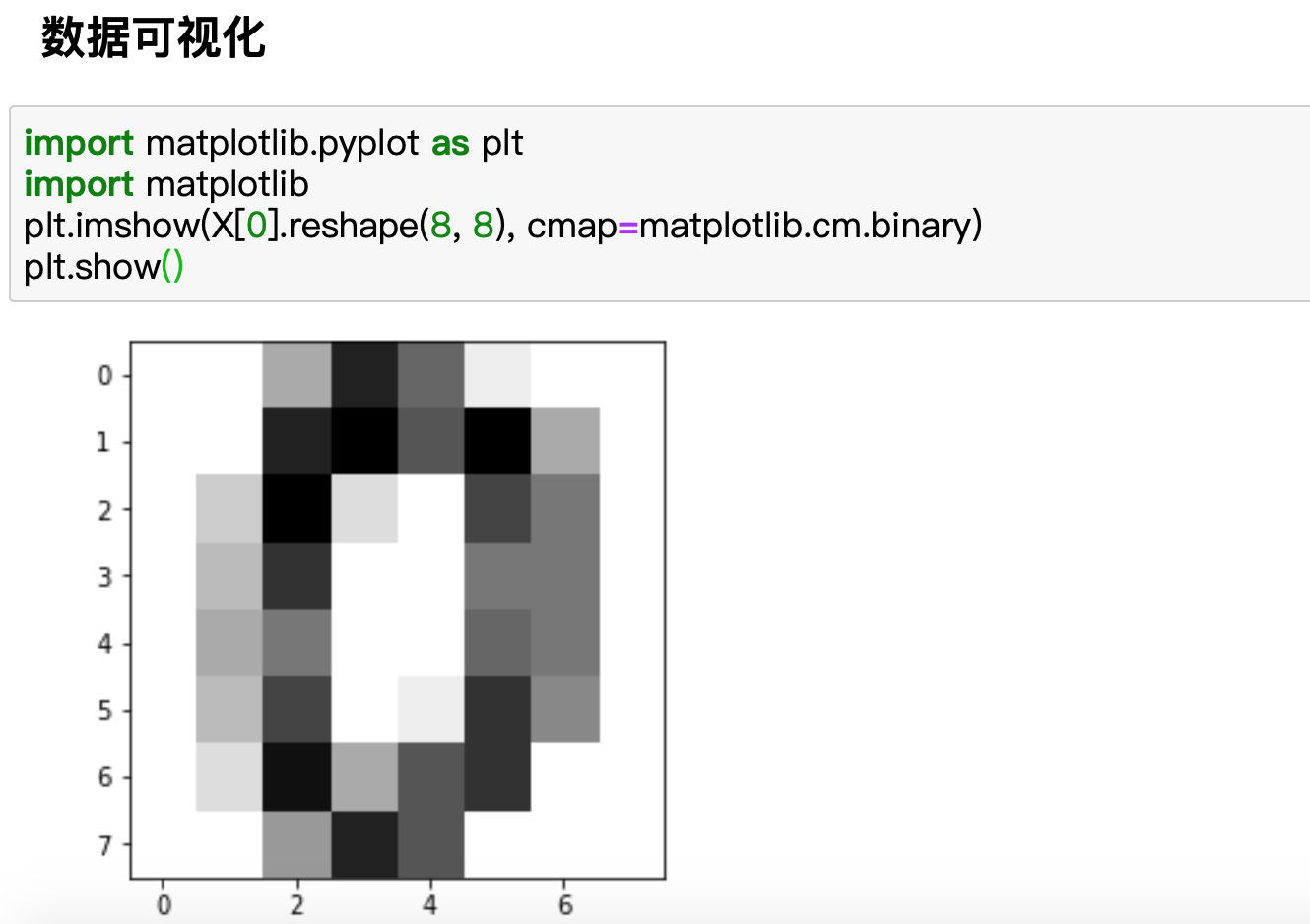
**如何寻找好的超参数?**

1. **业务知识**
2. **经验数值**
3. **实验测试**

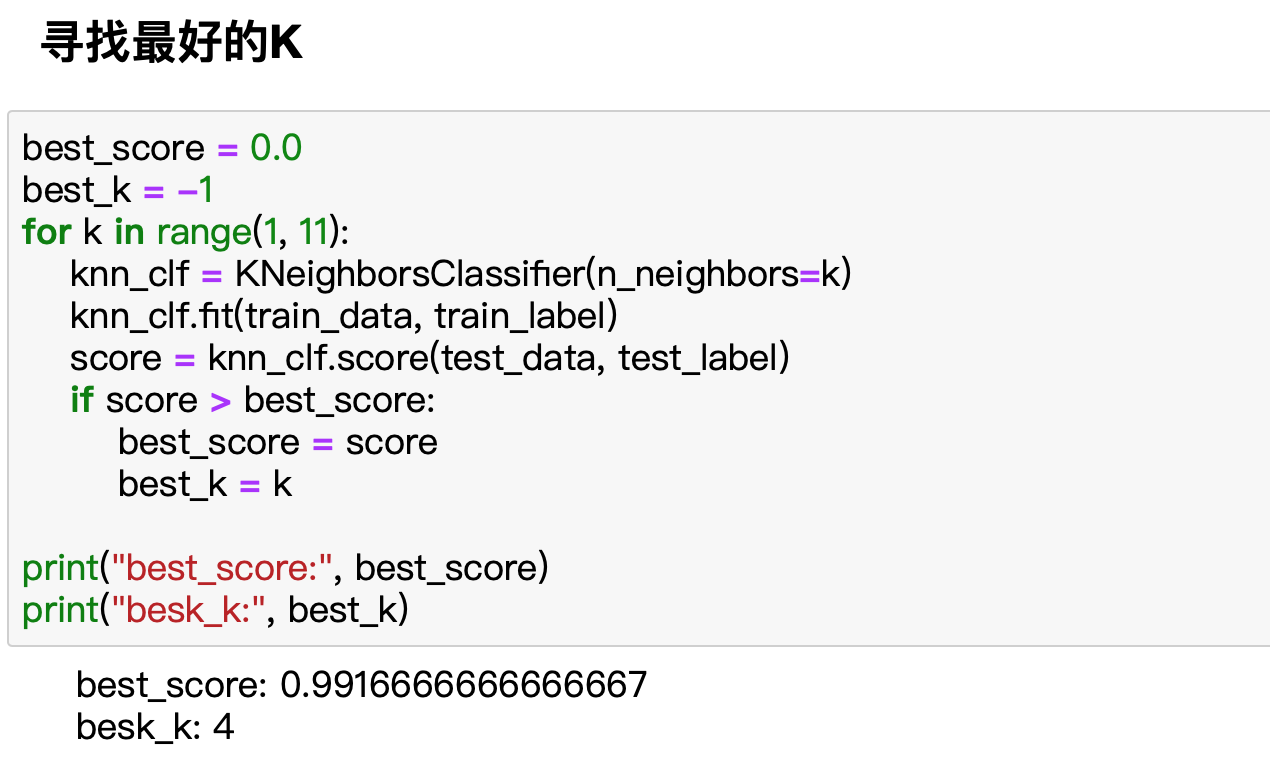
**超参数1：寻找最好的K值**

****

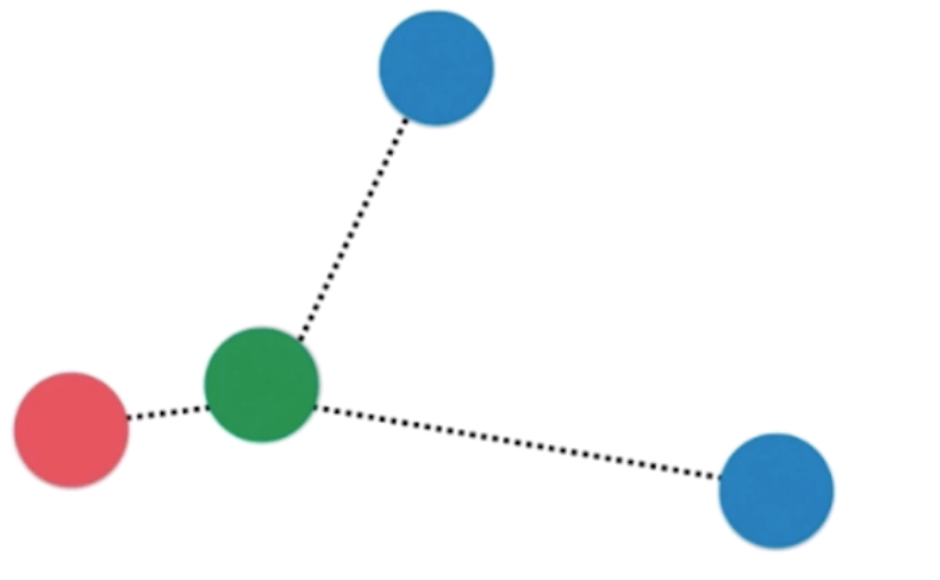
****

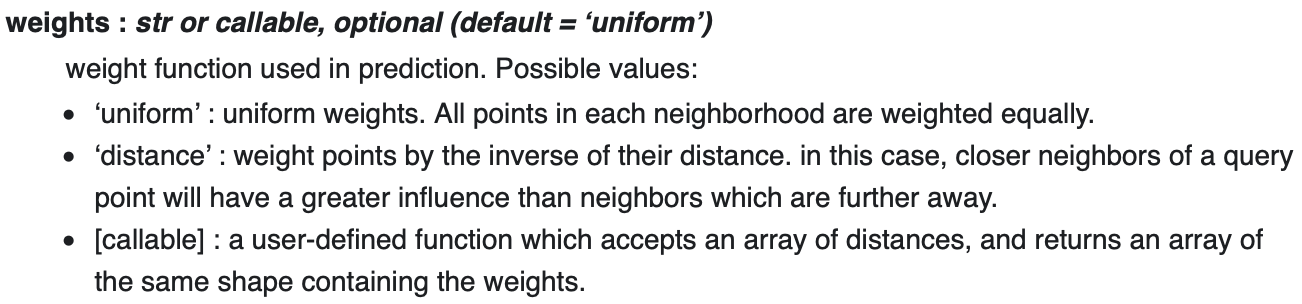
****

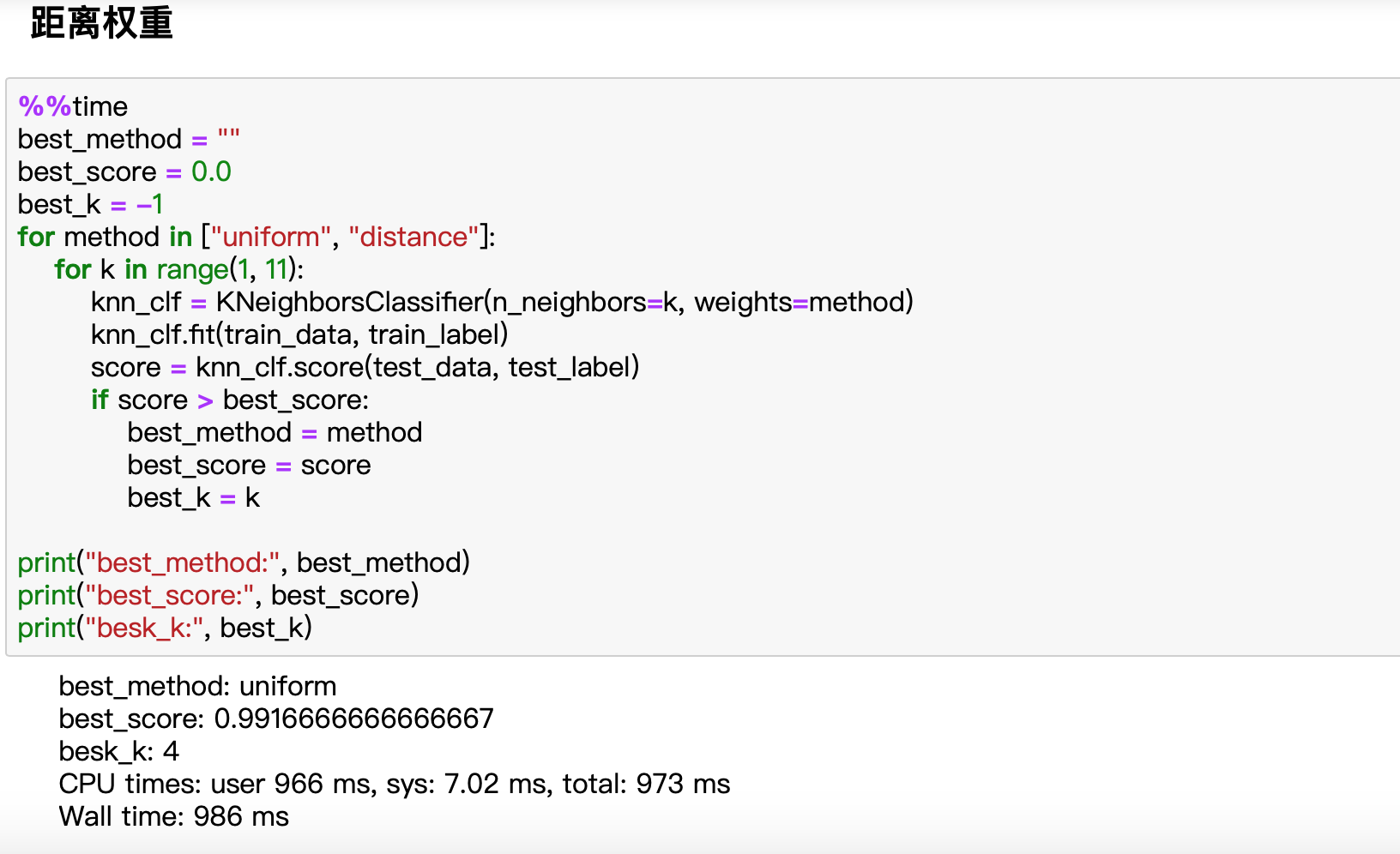
****

****

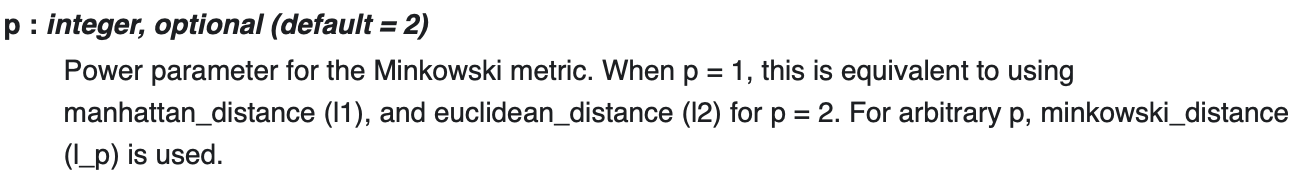
**超参数2：距离权重**

****

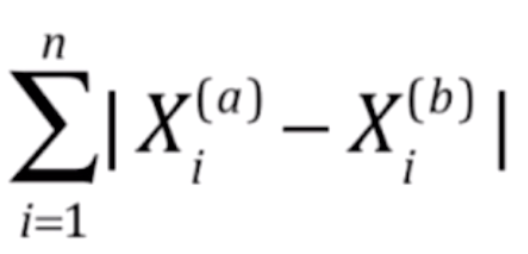
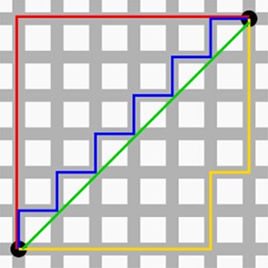
****

****

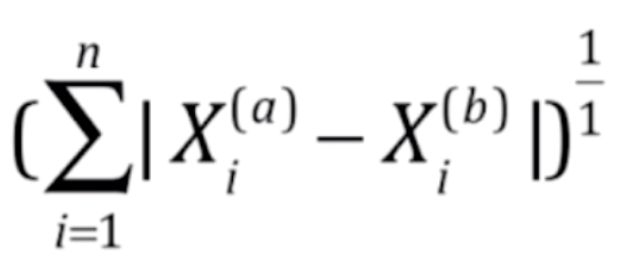
**超参数3：距离公式参数p**

****

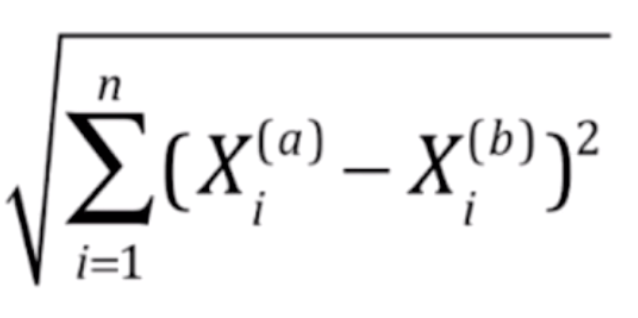
**1.曼哈顿距离**

**** 

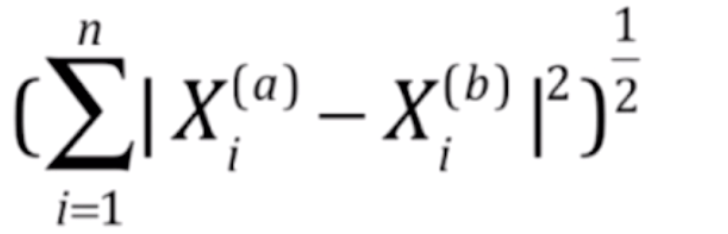
**曼哈顿距离公式变换**



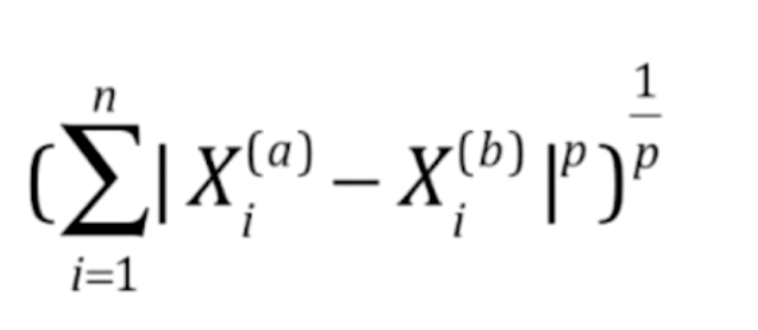
**2.欧氏距离**

****

**欧氏距离公式变换**

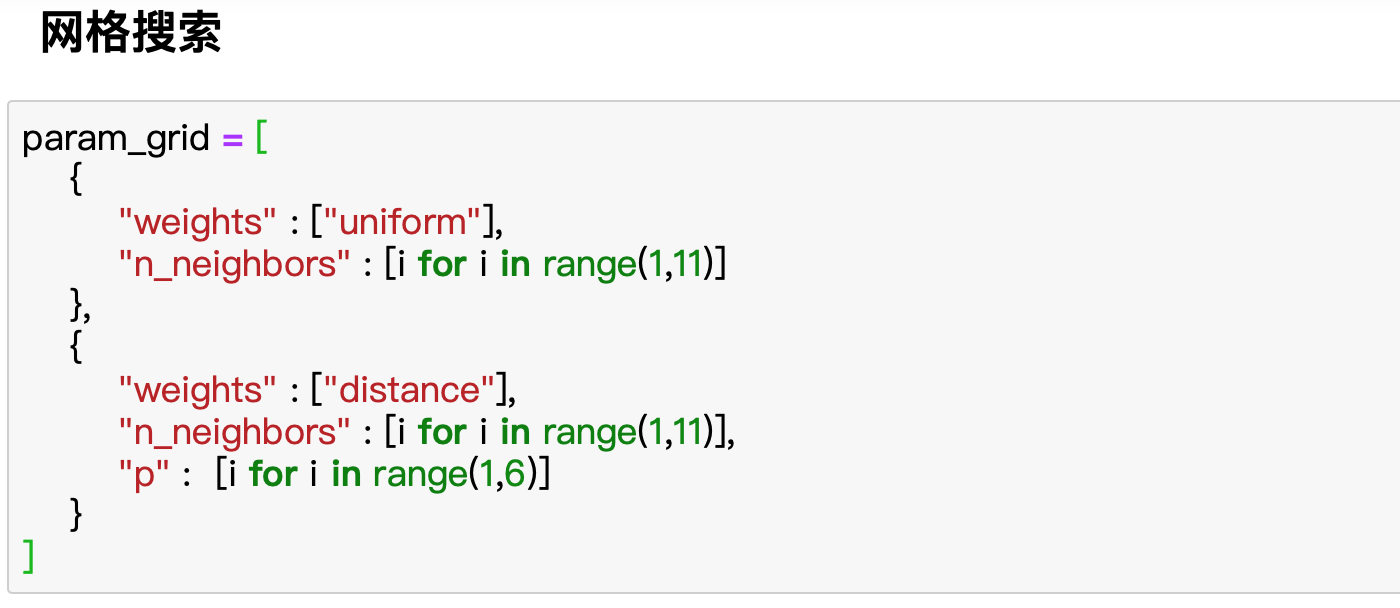
****

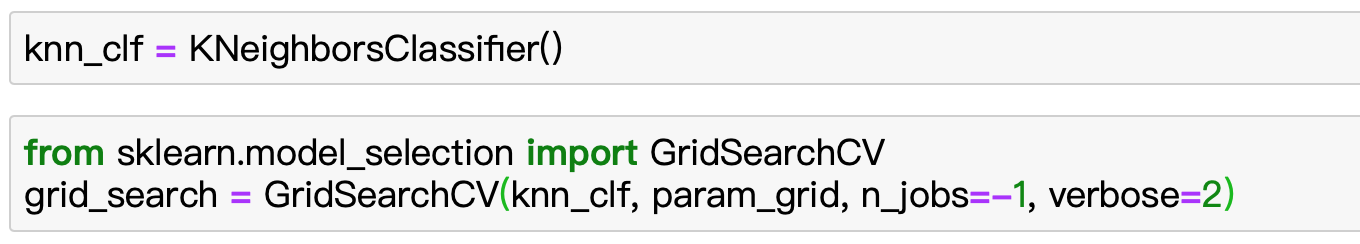
**3.明可夫斯基距离(minkowski\_distance)**

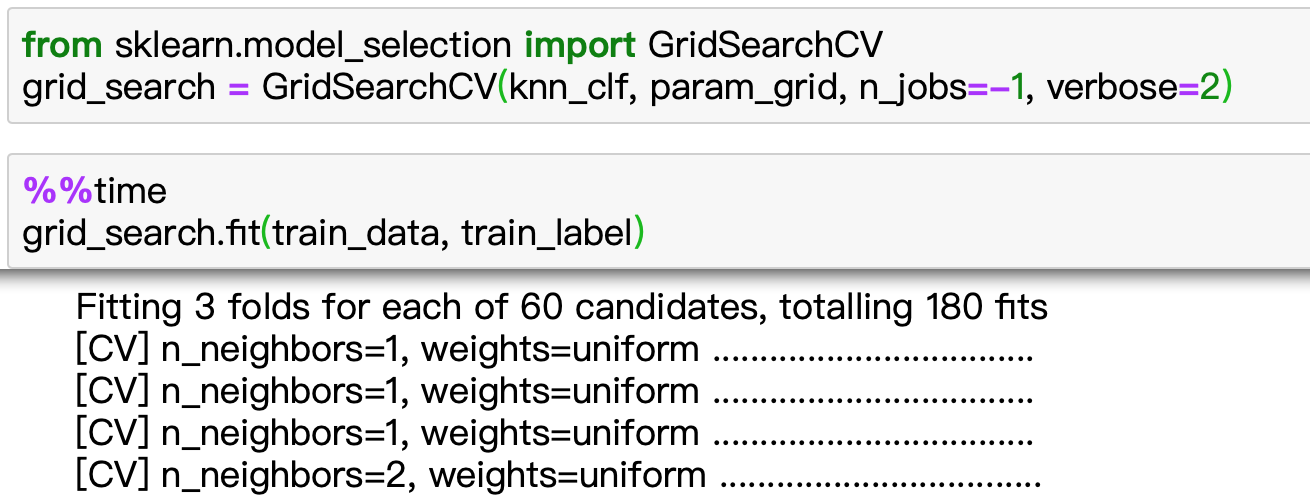
****

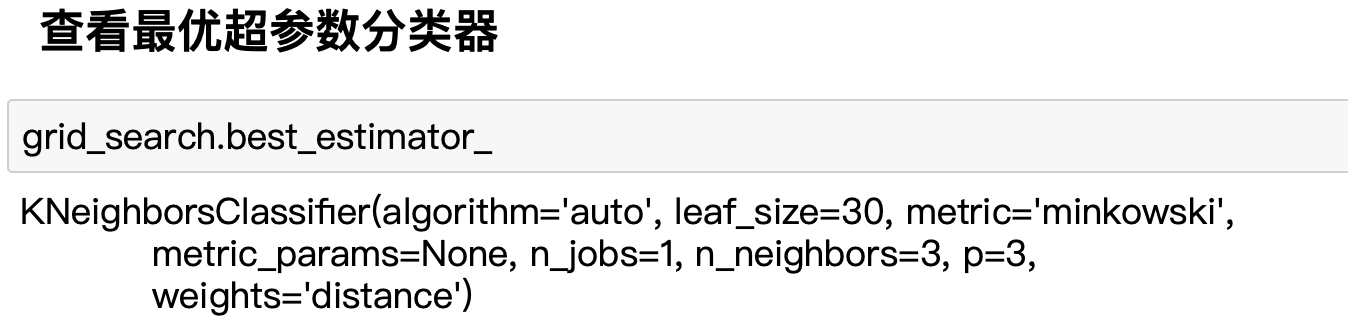
****

* 网格搜索

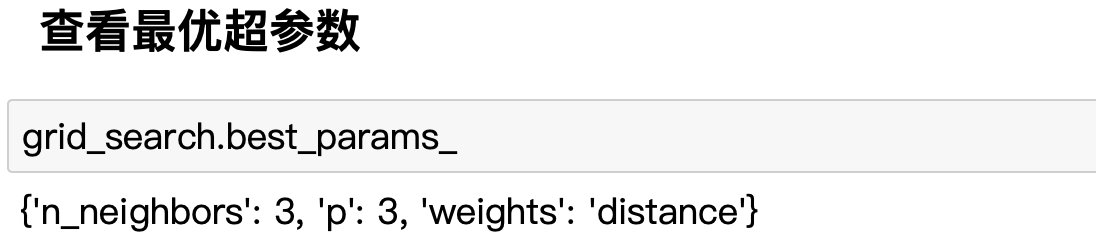












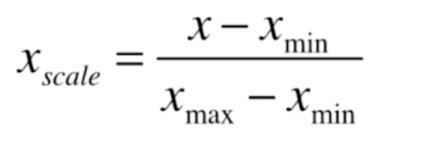


* 数据归一化

**归一化：将所有数据的量纲映射到同一尺度下。**

**归一化方式**

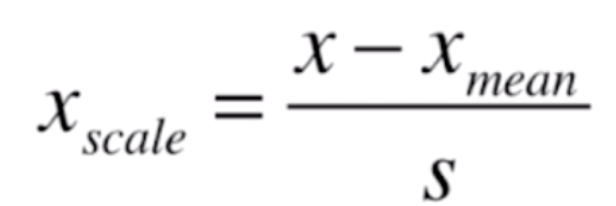
**1.最值归一化(normalization)：把所有的数据映射到0~1之间**

****

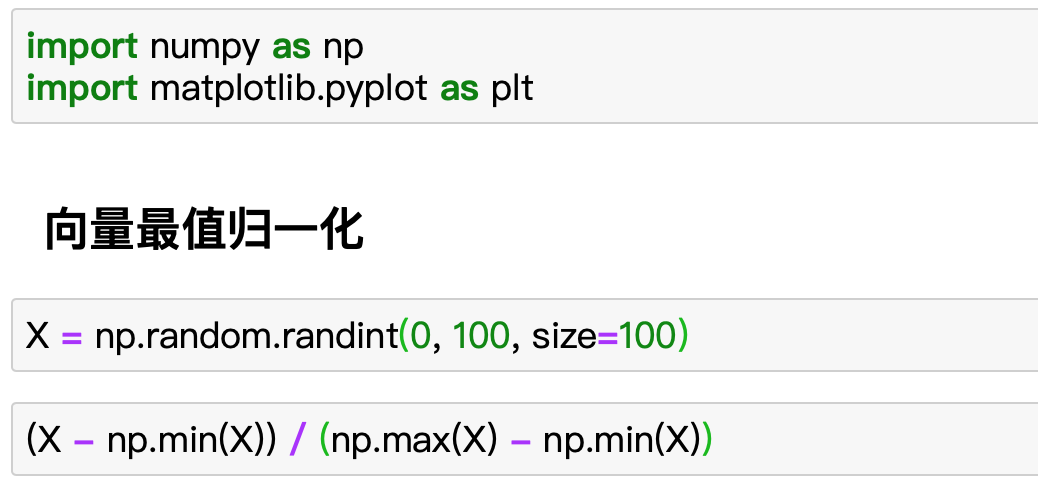
**适用场景：数据分布边界化，例如学生分数(最低0分，最高100分)**

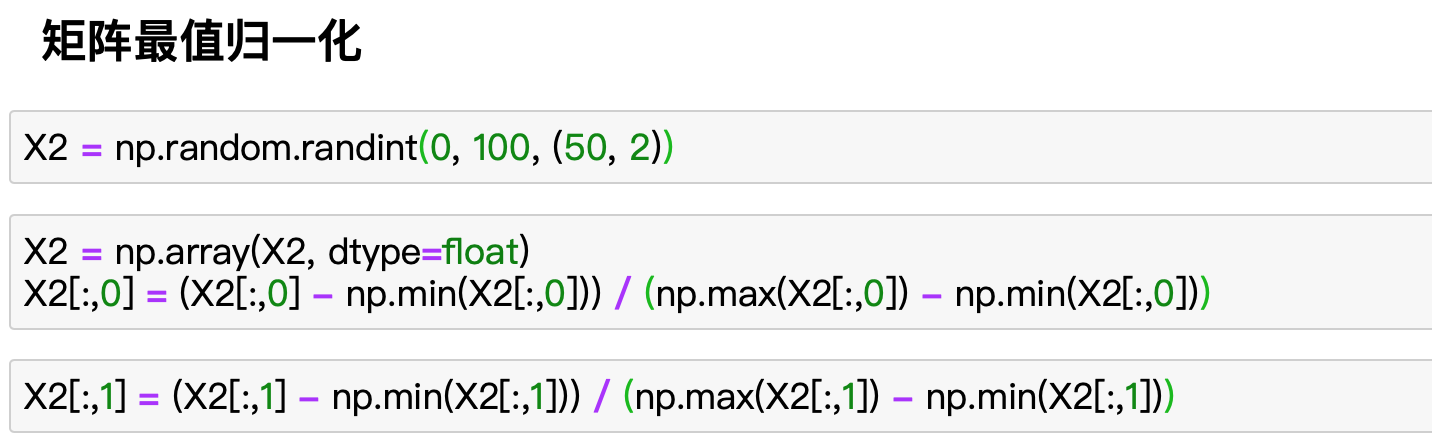
**缺点：受极值影响较大**

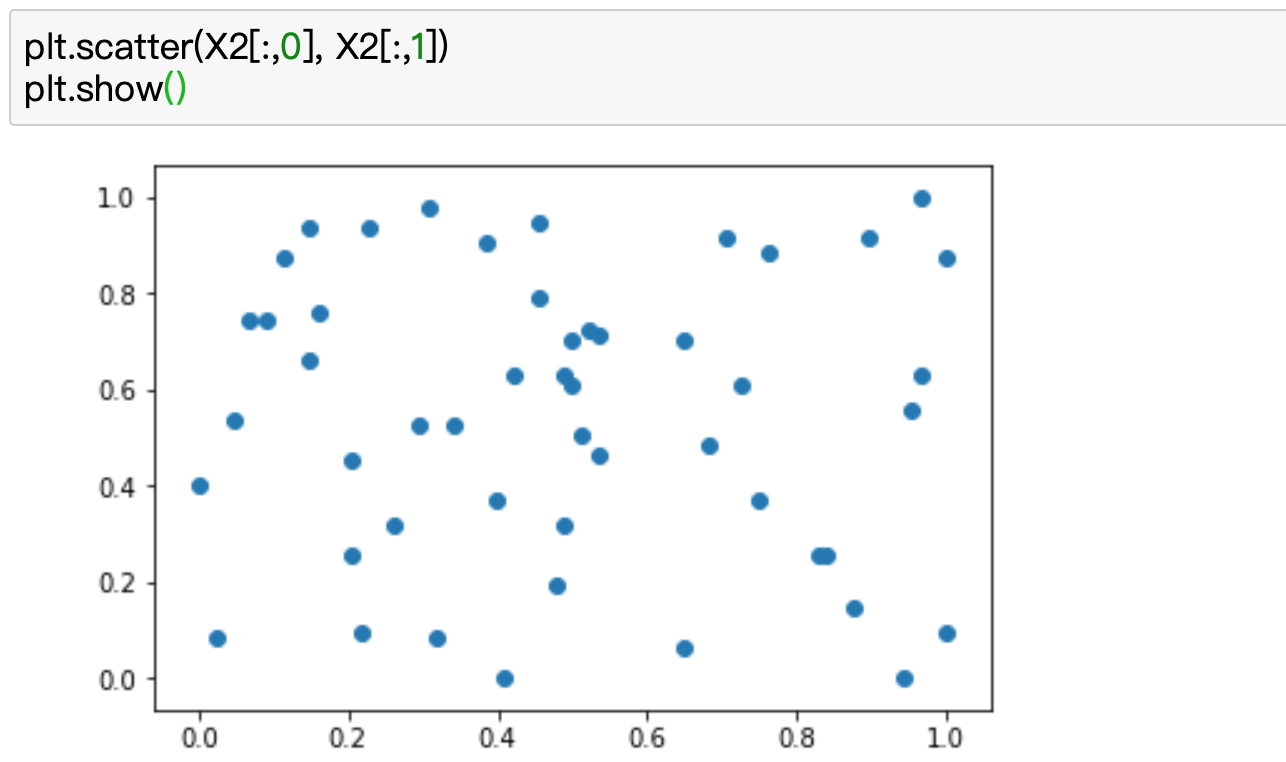
**2.均值方差归一化(standardization)：把所有的数据归一到均值为0方差为1的分布中**

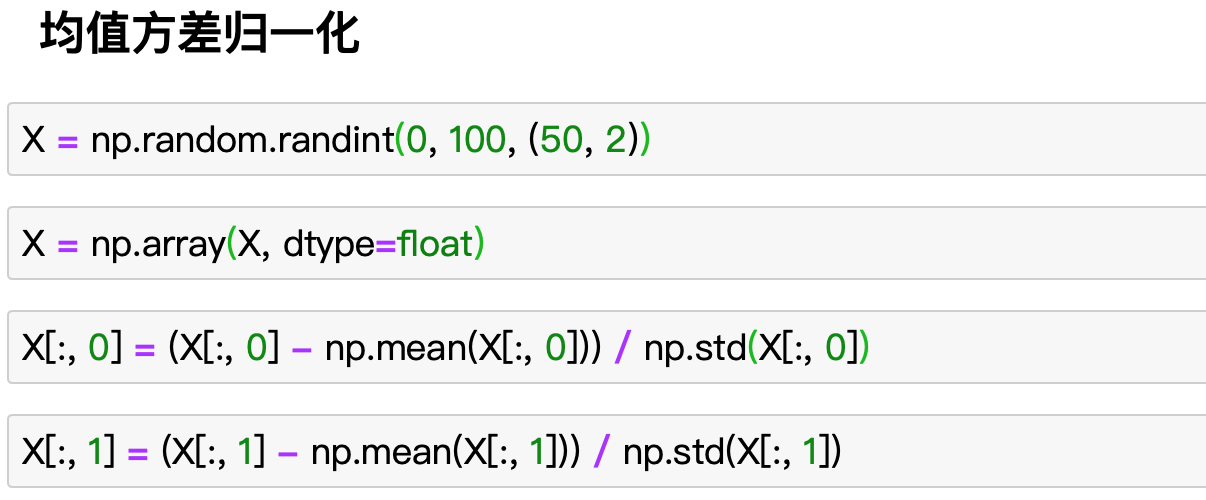
****

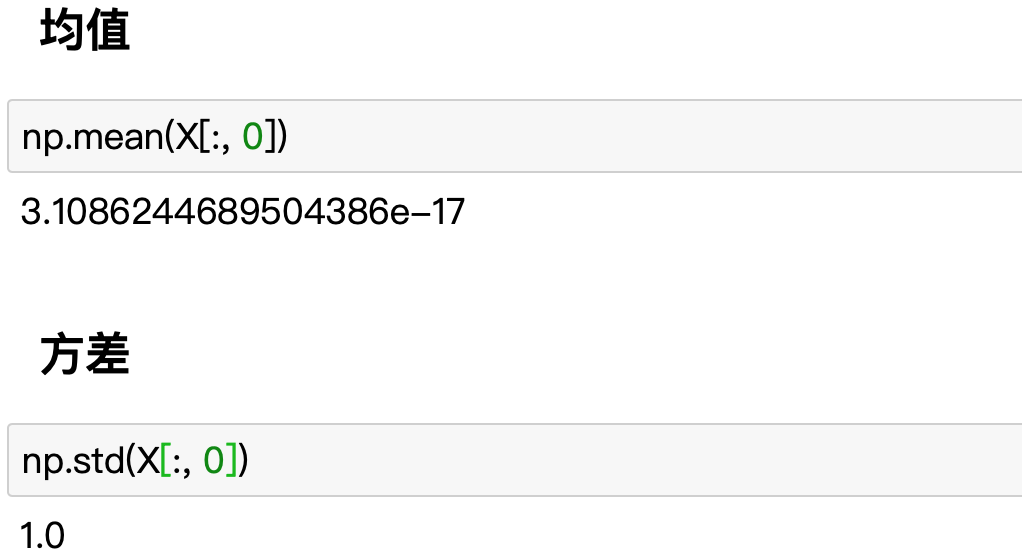
**适用场景：数据分布没有没有明确边界，有可能存在极端数据值**

****

****

****

****

****

* 如何而对真实数据进行归一化

**假设对数据进行均值方差归一化**

**可求出训练数据集的均值(mean\_train)和方差(std\_train)**

**如何求出测试集(x\_test)或真实数据的均值和方差?**

**测试集归一化**

**(x\_test - mean\_train) / std\_train**

**保存训练集得到的均值和方差**

**解决方案**

train data

input

Sklearn Scalar

fit

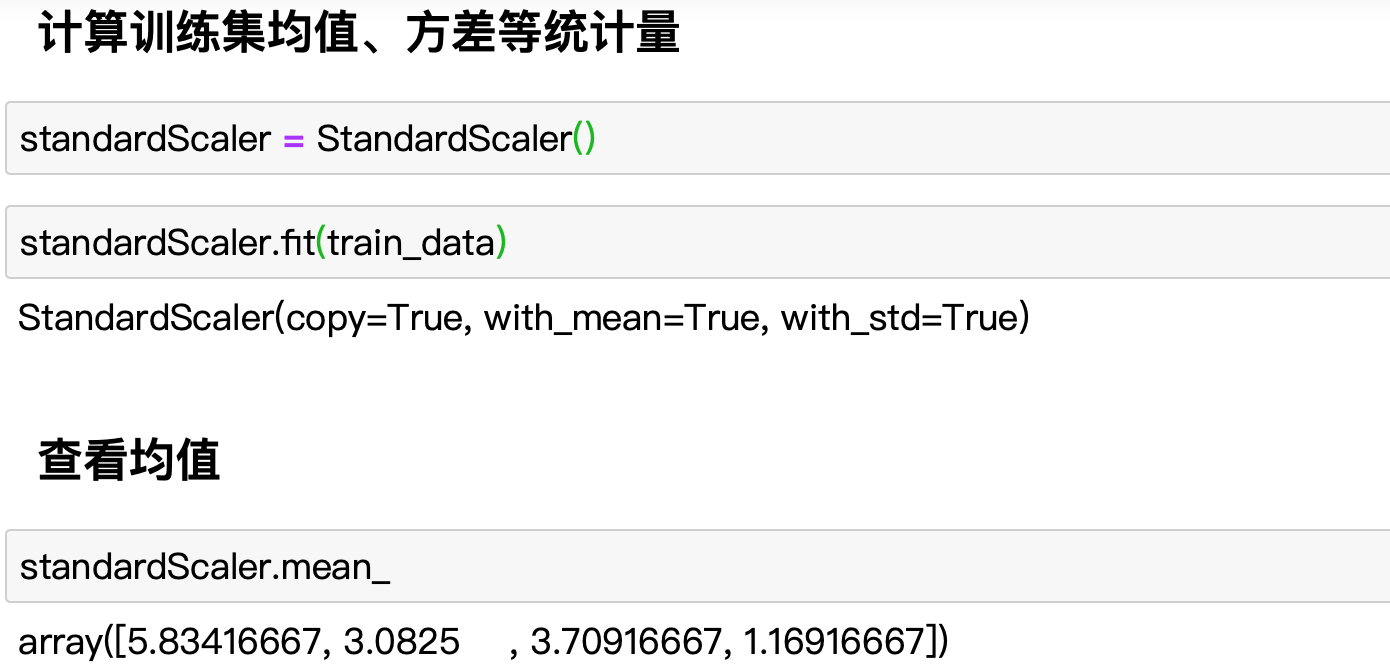
input

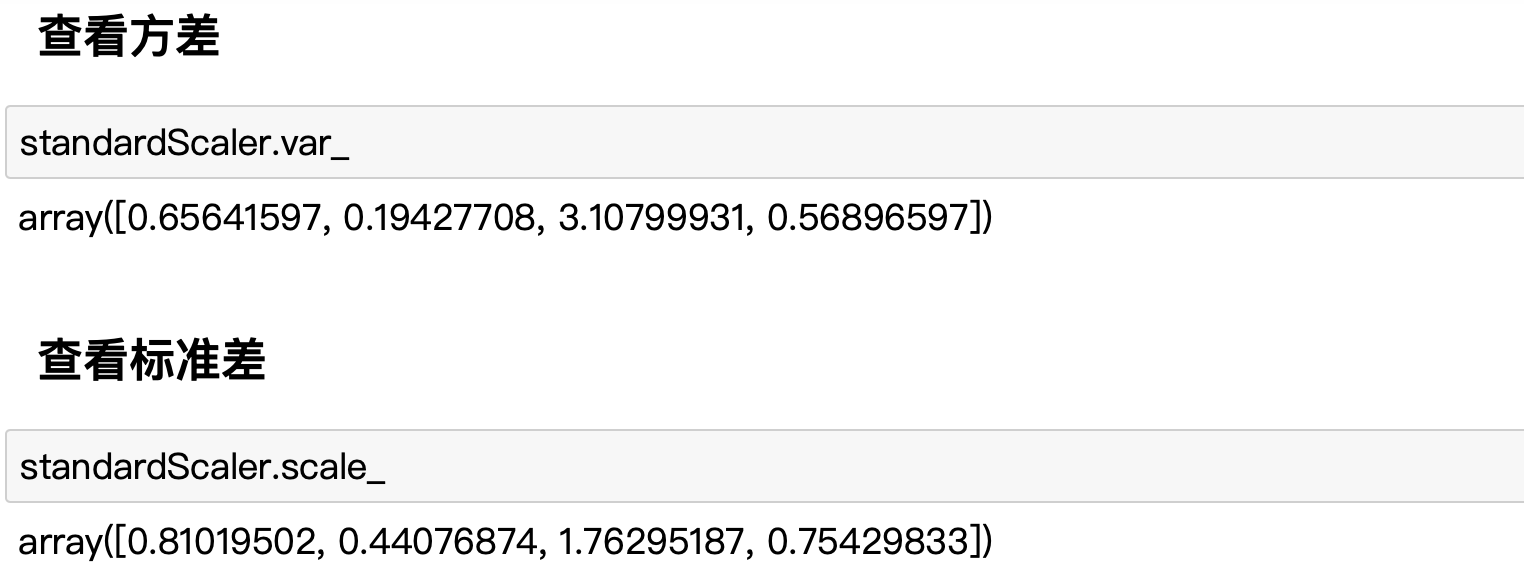
transform

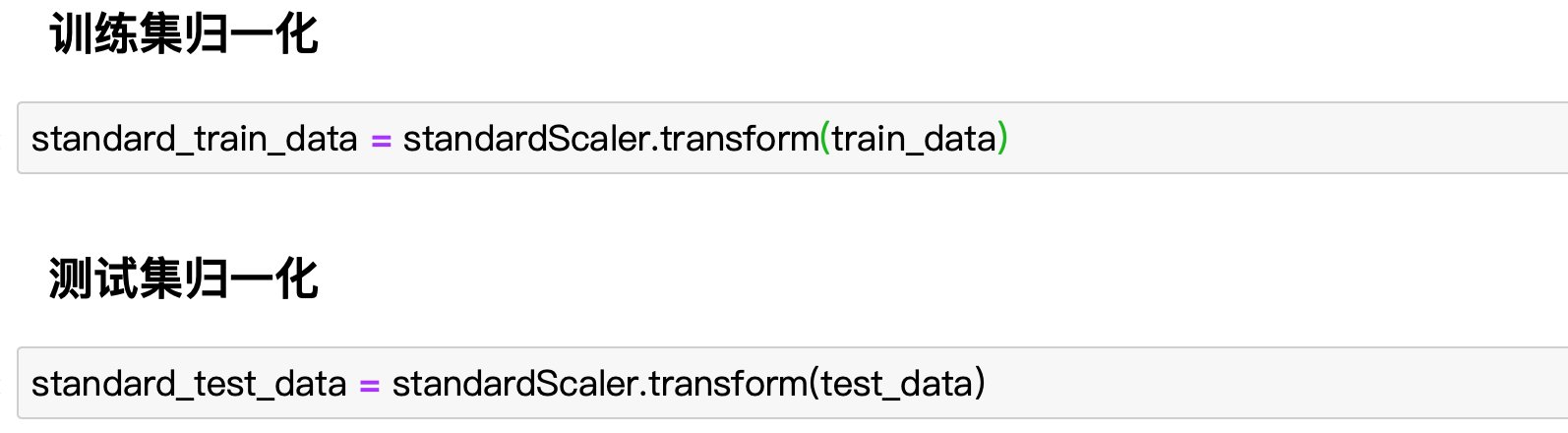
**归一化数据**

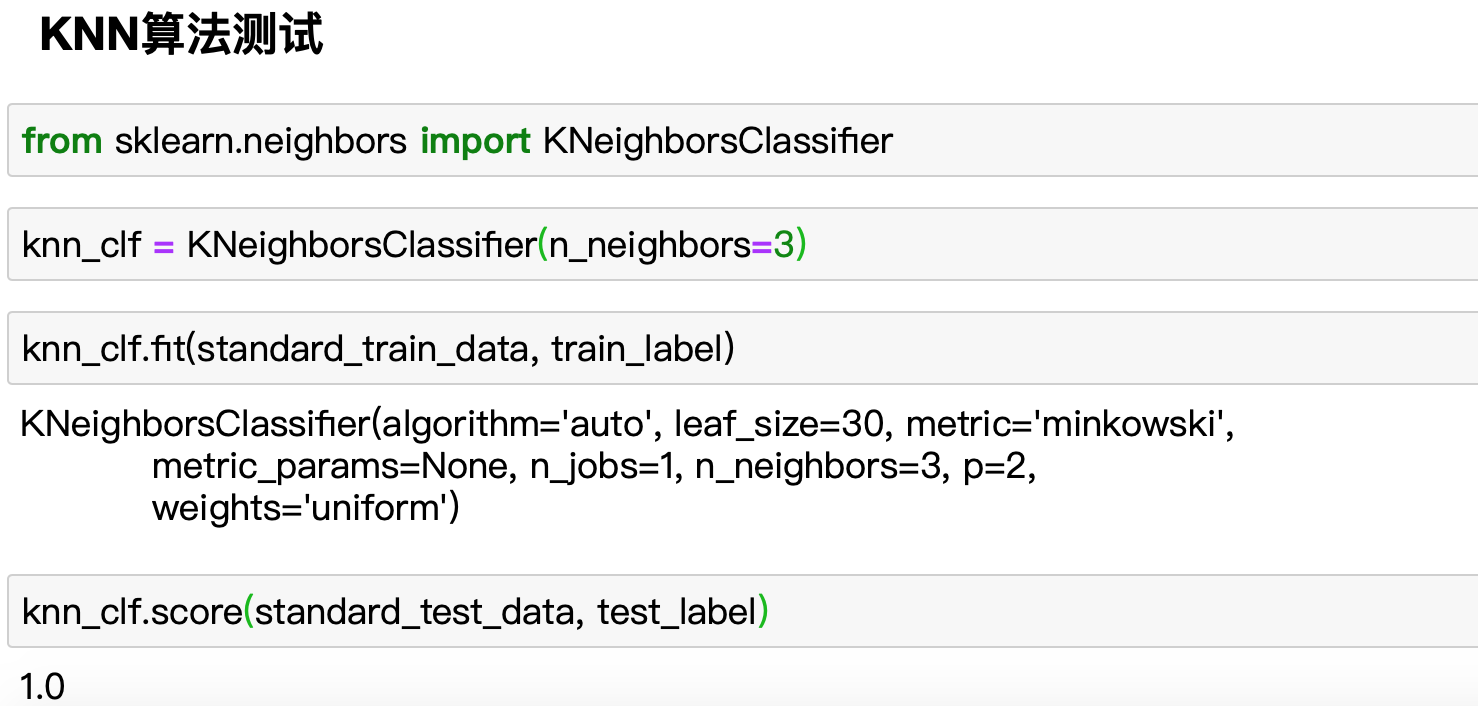
****

****

****

****

****

****