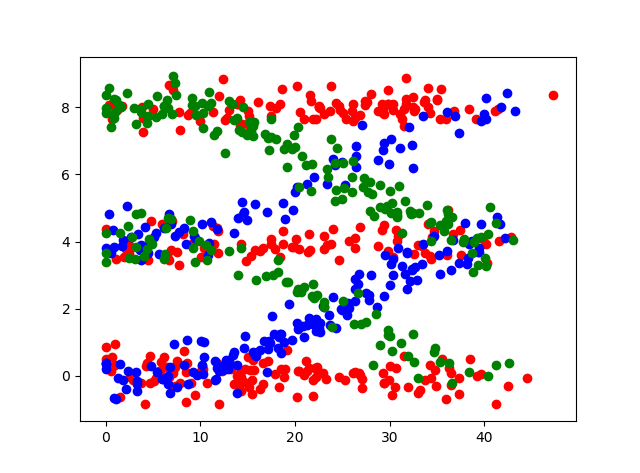
实现KNN分类器

在这个练习中，你将实现一个KNN分类器以预测公路上的车辆的行动。下图显示了有三个车道的公路上车辆的行为，其中，每个车道4米宽。点代表了车辆的坐标，d表示y坐标，s表示x坐标。车辆在公路上有三种行为：

1. 左拐（蓝色点）
2. 保持方向（红色点）
3. 右拐（绿色点）



给定车辆的坐标点，请写出KNN分类器对车辆的行为进行预测，也即，预测车辆是左拐、保持方向还是右拐。其中，一个坐标点包括以下四个值：

<s, d, s’, d’>

s和d含义如上所述；s‘和d’分别表示在x方向和y方向上的加速度。

**输入：**

有四个文件，分别是训练数据（包括参加训练的车辆的坐标点train\_states.txt和车辆的方向train\_label.txt）和测试数据(test\_states.txt和test\_label.txt)。

通过使用训练的坐标点数据进行训练，对测试数据进行预测。

**任务：**

1. 实现knn\_vehicle.py中的read\_data, read\_label，get\_Neighbors, get\_label, predict等函数。
2. 测试预测准确率。

**提示：**

可以参考课程中所演示代码。

另外，在这个例子中，需要认真选择样本点数据，也即，是否所有的数据都是有用的。

**代码：**

knn\_vehicle.py

**import** numpy **as** np  
**import** operator  
**from** numpy **import** \*

**#读取坐标点数据**

**def** read\_data(filename):  
 #待实现

**#读取坐标点对应的方向数据**

**def** read\_label(filename):  
 #待实现

trainingData = read\_data(**'train\_states.txt'**)  
trainingLabel = read\_label(**'train\_labels.txt'**)  
testData = read\_data(**'test\_states.txt'**)  
testLabel = read\_label(**'test\_labels.txt'**)  
 *#计算目标点和所有训练点之间的欧氏距离***def** euclid\_distance(x1, x2, dim):  
 distance = 0  
 sum = 0  
 **for** x **in** range(dim): *#遍历所有的训练点维度* distance = pow((x1[x] - x2[x]), 2)*#向量值平方* sum = sum + distance  
 *#print(sum)* **return** math.sqrt(sum)/2  
  
  
*# 计算所有目标点和所有训练点之间的距离，并选出K个最近邻***def** get\_Neighbors(training\_set, test\_set, k):  
 #待实现  
  
*# 对k个近邻进行合并，返回value最大的key***def** get\_label(index):  
 #待实现

#测试并统计正确率

**def** predict():  
 #待实现

#调用predict,显示预测结果

predict()