第一次竞赛文档

1. 实现思路
2. 收集数据：提供csv格式文件
3. 准备数据：编写函数read\_data()，读取训练数据和测试数据。
4. 分析数据：使用Python命令提示符中检查数据，确保它符合要求。
5. 训练算法：k-近邻算法
6. 测试算法：编写函数使用文件HTRU\_2\_test.csv中的数据作为测试样本，测试样本与非测试样本的区别在于测试样本是未知的类别，非测试样本是已经类别。
7. 使用算法
8. K近邻算法实现原理：对每一个未知点执行
9. 计算未知点到所有已知类别点的距离
10. 按距离排序（升序）
11. 选取其中前k个点中各个类别的个数
12. 上述k个点里类别出现频率最高的作为未知点的类别
13. 函数作用
14. read\_data() ：读取文件，返回训练数据、训练类别，测试数据

data：读取HTRU\_2\_train.csv文件，返回的是训练的数据与类别

test\_data：读取HTRU\_2\_test.csv文件，返回测试数据

利用切片方式对data进行切片，使数据和类别进行分隔

train\_data ：切片得到的训练数据

train\_label ：切片得到的训练类别

1. knn(trainData, testData, labels, k)：使用knn算法进行分类，返回出现频率最高的类别

参数介绍：

trainData：训练的数据

testData： 测试的数据

labels：训练数据对应的类别

k：选取的k值

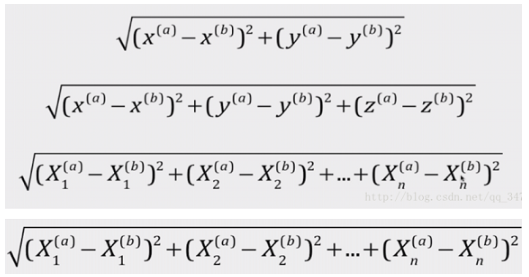
函数中用到变量介绍：

row ： 计算训练样本的行数

diff ： 计算训练样本的测试样本的差值

sqrDiff ：计算差值平方和

distances：计算到各个样本点的距离，利用欧氏距离



sortDistance：对距离进行从高到低的排序

sortCount：对出现的频率从高到低进行排序

1. main() ：主函数

调用read\_data() 进行数据的读取，获取测试数据、训练数据、训练类别。

根据函数knn() 返回的频率最高的类别做为预测类别。

将测试数据的行号和列号写进knn35.csv文件中。