《Python语言与编程》课程实验指导书

**实验三 Python应用 (人工智能，网络通信)**

1. 实验目的

学生通过使用Python进行机器学习K近邻算法实践与网络编程实践，理解并掌握Python解决实际问题的方法，同时巩固加深对已学Python的知识的理解与掌握。

1. 实验内容

（详细内容见附件一）

1. 实验环境

Windows操作系统，Python 3.6

1. 实验要求
2. 独立完成实验内容要求。
3. 使用Python的IDE进行编程，shell下演示结果（截屏保存）。
4. 需要上交源程序文件（以题目为编号）。
5. 实验学时

2学时

**附件一**

**1 KNN算法**

**现有二维空间的样本点（如：p(x,y)）,样本分为两类(类别1，类别2)。现提供了200个数据样本及其标签供使用。有10个未知样本待分类。**

1. **编程实现KNN算法。**
2. **对10个未知样本点进行分类。**
3. **请尝试几个不同的K值（至少要包含K=1的情况），比较结果。**

**2简单的数据收发器**

1. **分别编写数据发送程序，数据接收程序。**
2. **基于UDP协议实现。**
3. **发送端发送的信息包括：发送时间，IP，端口，消息内容。**
4. **接收端在接收到信息后将其显示出来，并保存在一个文件中。如果接收到“Bye”或“bye”，则退出。**

附件二：

**北京邮电大学软件学院**

**2017－2018学年第二学期实验报告**

**课程名称： Python语言与编程**

**项目名称： Python编程（应用练习）**

**项目完成人：**

**姓名：\_\_\_\_**苏可欣**\_\_\_\_学号：\_\_\_\_\_2016211954\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**指导教师：\_\_\_\_管皓\_\_\_\_\_\_\_\_\_\_\_\_**

**日 期： 2018 年6 月20 日**

1. 实验目的

学生通过使用Python进行机器学习K近邻算法实践与网络编程实践，理解并掌握Python解决实际问题的方法，同时巩固加深对已学Python的知识的理解与掌握。

1. 实验内容

**1 KNN算法**

**现有二维空间的样本点（如：p(x,y)）,样本分为两类(类别1，类别2)。现提供了200个数据样本及其标签供使用。有10个未知样本待分类。**

1. **编程实现KNN算法。**
2. **对10个未知样本点进行分类。**
3. **请尝试几个不同的K值（至少要包含K=1的情况），比较结果。**

**2简单的数据收发器**

1. **分别编写数据发送程序，数据接收程序。**
2. **基于UDP协议实现。**
3. **发送端发送的信息包括：发送时间，IP，端口，消息内容。**
4. **接收端在接收到信息后将其显示出来，并保存在一个文件中。如果接收到“Bye”或“bye”，则退出。**
5. 实验环境

Windows操作系统，Python 3.6

1. 实验要求
2. 独立完成实验内容要求。
3. 使用Python的IDE进行编程，shell下演示结果（截屏保存）。
4. 需要上交源程序文件（以题目为编号）。

**五. 实验结果**

（说明实验完成情况）

**knn**算法的实现过程如下：

**1.** 计算已知类别数据集中的点与当前点之间的距离（欧式距离）

**2.** 按照距离递增次序排序

**3.** 选取与当前点距离最小的**K**个点

**4.** 确定前**K**个点所在类别的出现频率

**5.** 返回前**k**个点出现频率最高的类别最为当前点的预测分类

**6. k**表示用于选择最近邻的数目

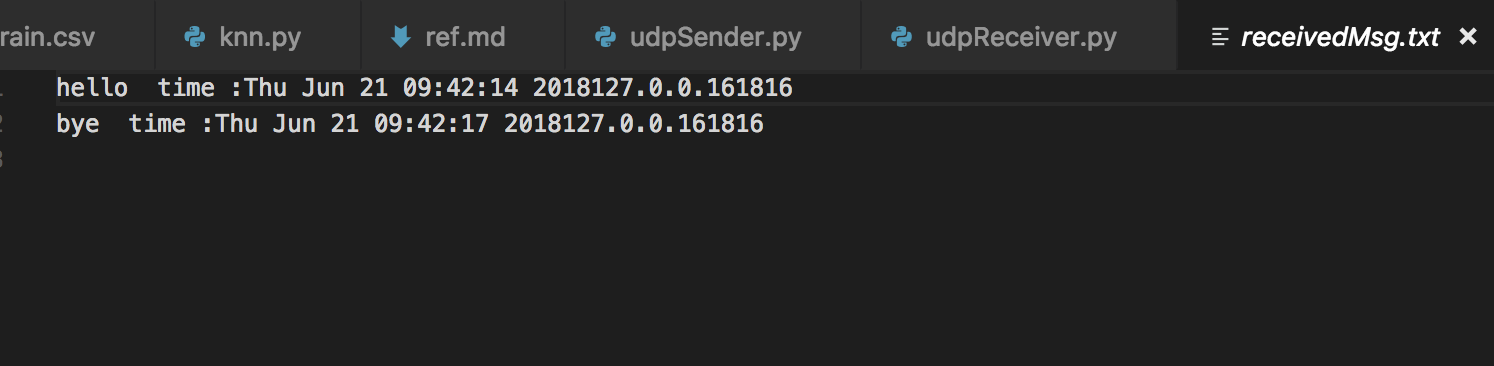
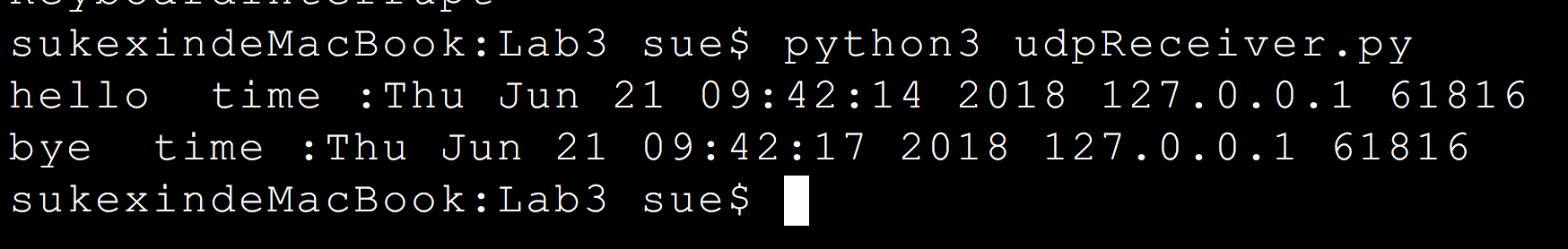
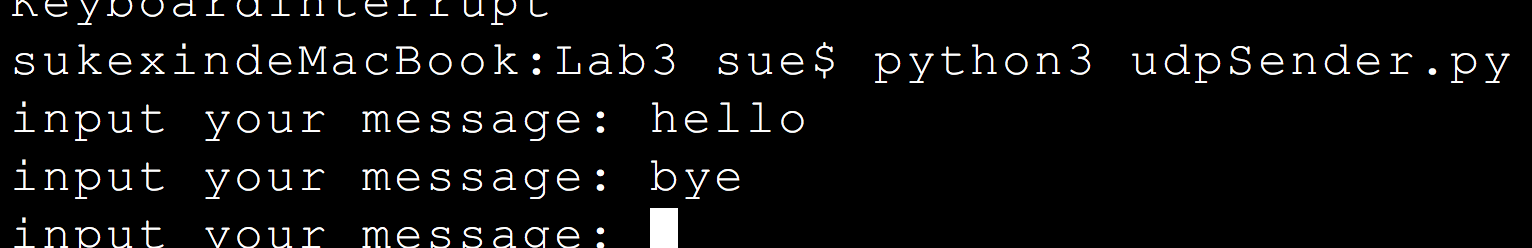
由下图的实验截图不难看到，**k**值的选择对预测结果有比较大的影响。

分析结果如下：

如果K值过小，相当于只用与输入实例较近（相似）的样本会对预测结果产生影响，这样带来的不足之处是预测结果会对近邻的样本点非常敏感，如果邻近的样本点恰好是噪声点，那么预测结果就会出错。换句话说，**K**值的减小意味着模型变得复杂，容易发生过拟合。

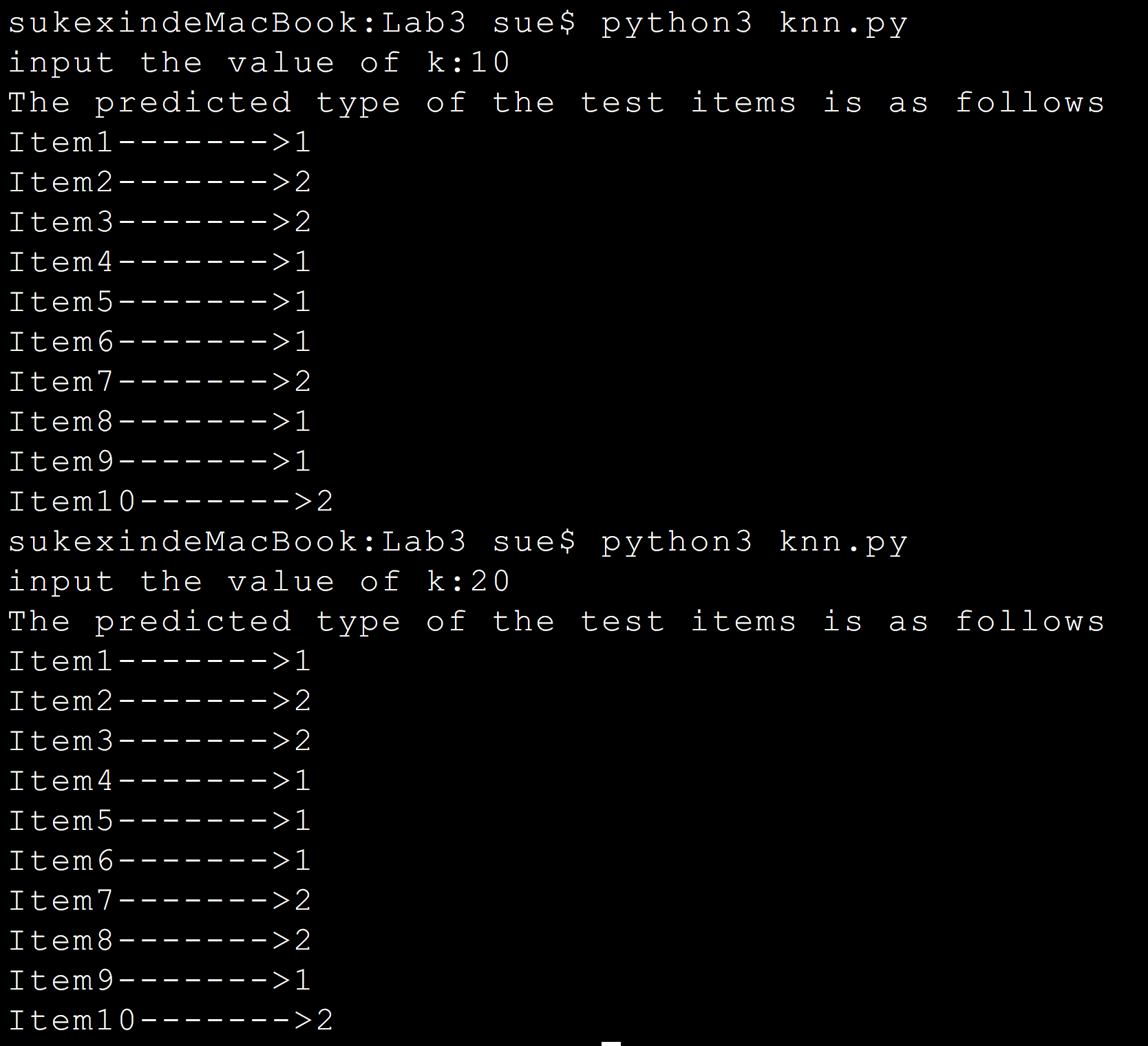
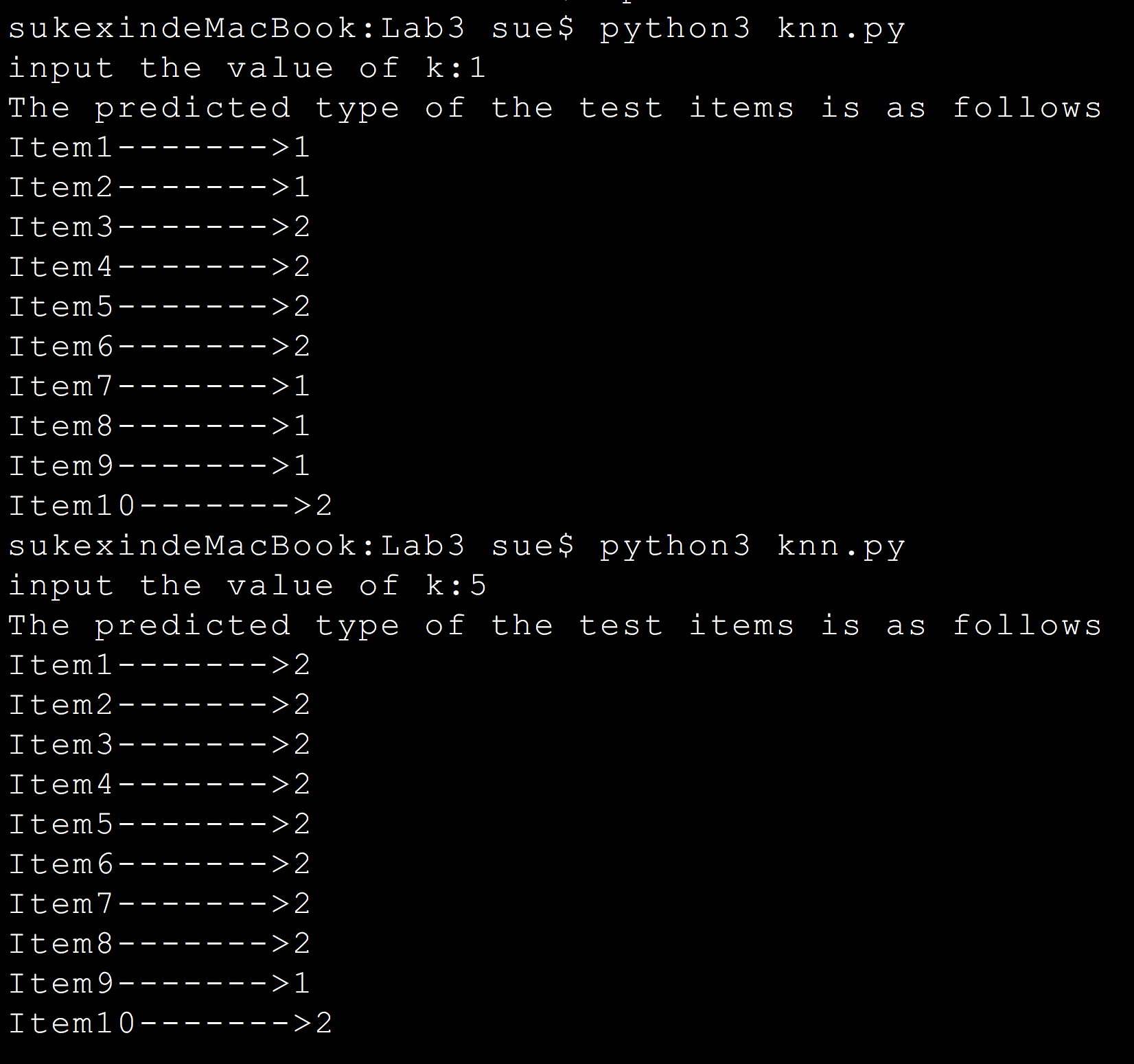
相反，如果**K**值过大，那么预测结果取决于输入实例较大邻域内的样本点，这样带来的不足之处是预测结果可能受到较远（不太相似）的样本点的干扰，从而使得预测结果不准确。

换句话说，**K**值增大意味着模型变得简单，极端情况下，K=N，那么无论输入实例是什么，都将简单地预测它属于训练实例中出现最多的类，这时，模型过于简单，完全忽略了训练集中有用的信息，所以该方案是不可取的。 通常，***K***值一般取一个较小的数值，然后使用交叉验证法来选取最优的***K***值



实验二截图

在TCP/IP模型中，UDP为[网络层](https://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E5%B1%82)以上和[应用层](https://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E5%B1%82)以下提供了一个简单的接口。UDP只提供数据的不可靠传递，它一旦把应用程序发给网络层的数据发送出去，就不保留数据备份（所以UDP有时候也被认为是不可靠的数据报协议）。UDP在IP数据报的头部仅仅加入了复用和数据校验（字段）。

UDP首部字段由4个部分组成，其中两个是可选的。各16[bit](https://zh.wikipedia.org/wiki/%E4%BD%8D%E5%85%83)的来源端口和目的端口用来标记发送和接受的应用进程。因为UDP不需要应答，所以来源端口是可选的，如果来源端口不用，那么置为零。在目的端口后面是长度固定的以字节为单位的长度域，用来指定UDP数据报包括数据部分的长度，长度最小值为8byte。首部剩下地16bit是用来对首部和数据部分一起做[校验和](https://zh.wikipedia.org/wiki/%E6%A0%A1%E9%AA%8C%E5%92%8C)（Checksum）的，这部分是可选的，但在实际应用中一般都使用这一功能。

由于缺乏可靠性且属于非连接导向协议，UDP应用一般必须允许一定量的丢包、出错和复制粘贴。但有些应用，比如[TFTP](https://zh.wikipedia.org/wiki/TFTP)，如果需要则必须在应用层增加根本的可靠机制。但是绝大多数UDP应用都不需要可靠机制，甚至可能因为引入可靠机制而降低性能。[流媒体](https://zh.wikipedia.org/wiki/%E6%B5%81%E5%AA%92%E9%AB%94)（流技术）、即时多媒体游戏和[IP电话](https://zh.wikipedia.org/wiki/IP%E7%94%B5%E8%AF%9D)（VoIP）一定就是典型的UDP应用。如果某个应用需要很高的可靠性，那么可以用[传输控制协议](https://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)（TCP协议）来代替UDP。