第二次竞赛说明文档

一、题目和数据的简单分析

首先分析竞赛题目，是有关于收入水平级别（1-9）的多分类问题，训练集（test.csv）中有7193行，14列数据，测试集（train.csv）中有1798行，13列数据。训练集和测试集中含有“缺失值”（？）。解决多分类的方法有很多，比如：KNN算法，决策树算法，朴素贝叶斯算法，softmax回归算法等，我选择的是KNN算法。

二、KNN算法简单分析

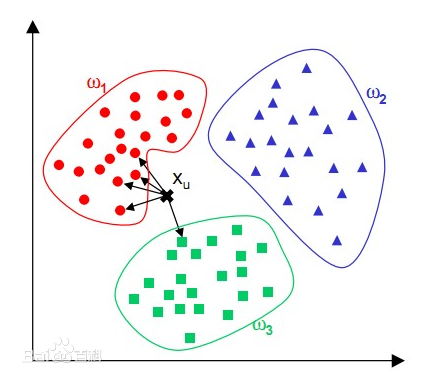
**1**.KNN是一种**非参的**，**惰性**的算法模型。KNN是最简单的分类算法之一，同时也是最常用的算法之一。

**2.KNN的原理就是当预测一个新的值x的时候，根据它距离最近的K个点是什么类别来判断x属于哪个类别，**通过度量”待分类数据”和”类别已知的样本”的**距离**（通常是欧氏距离）对样本进行分类。

**3**.KNN算法流程

对每一个未知点执行：

1. 计算未知点到所有已知类别点的距离
2. 按距离排序（升序）
3. 选取其中前k个与未知点离得最近的点
4. 统计k个点中各个类别的个数
5. 上述k个点里类别出现频率最高的作为未知点的类别



三、代码分析

代码的实现共引用了三个库：Pandas、NumPy、sklearn

1.首先读取训练集，对.csv文件进行读取。

pd.read\_csv('train.csv',header=-1)

因为数据没有表头标题，所以要加header条件。

训练集中的“缺失值”用“？”来代表的，所以首先应将“？”变成真正的缺失值。我用的是replace()的方法，将“？”变成“np.NaN”。

2.训练集中的缺失值真正存在之后，以每列为单位进行缺失值数目的查看。这里用的是info()方法。

可以看出训练集中并不存在很大面积的缺失问题。

3.对训练集进行缺失值补全，并进行再一次的检验。

这里用到的缺失值补全是方法是：用前一个数据填充。

fillna(method='pad')

再一次检验训练集，发现缺失值都已补全。

4.重复上面的步骤，将测试集进行导入、缺失值补全

5.用KNN算法进行分类

选取训练集的前13列作为训练集数据，最后一列为训练标签。

X\_train = train.iloc[:,:-1]

y\_train = train[13]

这里选的n\_neighbors=43，通过了对KNN参数的优化得出的。

然后对测试集进行测试，得到测试类别结果。

6.将得到的分类结果导入到predict.csv文件中去，并添加一列序号和一行标题。

四、代码编写过程中遇到的问题和解决变法与代码的不足

对缺失值的理解不够深刻，在读取缺失值时耗费了一定的时间，通过查资料明白“”、NaN、null、none等的区别。

在对测试集进行缺失值补全时，第9列数据在检测时仍然存在缺失值，查看数据之后发现第9列的第一个数据是缺失值，故第9列数据用后一个数据进行的缺失值补全。

KNN参数的优化处理的不是很好，还需要进行改进。

可以在后续中选用其他分类方法对数据进行分类，多进行尝试，加深理解。