实验三：使用K-近邻模型实现空气质量的预测

主讲教师：严资林

实验教师：匡慈维

目录

本学期实验总体安排

本学期实验课程共 10 个学时，5 个实验项目，总成绩为 20 分。

实验项目	一	二	三	四	五
学时	2	2	2	2	2
实验内容	感知机模型	决策树模型	K近邻模型	支持向量机模型	聚类模型
分数	3	4	4	4	5
上课时间(地点)	第11周 周四 (T2102)	第12周 周六 (T2102)	第14周 周四 (T2102)	第16周 周二 (T2102)	第17周 周四 (T2102)
检查方式	提交实验截图文档	提交实验报告、工程文件			

5-6节 3&4班； 7-8节 1&2班

线上腾讯会议：848-8762-6539

实验任务

空气污染是一个复杂现象，在特定时间和地点，空气污染浓度会受许多因素的影响。目前，参与空气质量等级评定的主要污染物包含细颗粒PM2.5、可吸入颗粒物PM10、SO2、CO、NO2、O3等等，现需要构建一个**K近邻**模型，预测其质量等级。

- **任务一**：使用Python自编程构建K近邻模型，实现空气质量的预测与评价。
- **附加题**：使用sklearn中K近邻模型，对空气质量数据进行预测分类与评价。（选做）

注：附加内容有10%的加分， 但总分不超过该次实验满分。

数据说明

◆ 数据集

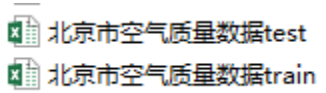
- 包含**训练集train**（共1725条数据），**测试集test**（430条数据）
- 每一条数据由 7 个特征值及1个目标值组成。
- 7 个特征值分别为：

  日期、PM2.5、PM10、SO2、CO、NO2、O3
- 目标值为6种不同类别的空气质量等级，分别为:

  优、良、

  轻度污染、中度污染、

  严重污染、重度污染



	A	B	C	D	E	F	G	H
1	日期	PM2.5	PM10	SO2	CO	NO2	O3	质量等级
2	2014/1/1	45	111	28	1.5	62	52	良
3	2014/1/2	111	168	69	3.4	93	14	轻度污染
4	2014/1/3	47	98	29	1.3	52	56	良
5	2014/1/4	114	147	40	2.8	75	14	轻度污染
6	2014/1/5	91	117	36	2.3	67	44	轻度污染
7	2014/1/6	138	158	46	2.4	68	12	中度污染
8	2014/1/7	111	125	34	2	60	43	轻度污染
9	2014/1/8	15	25	13	0.5	21	53	优
10	2014/1/9	27	46	19	0.8	35	53	优
11	2014/1/10	63	94	53	1.9	71	19	良
12	2014/1/11	106	128	76	2.8	90	11	轻度污染
13	2014/1/12	27	47	27	0.7	39	59	优
14	2014/1/13	82	107	67	2.3	78	20	轻度污染
15	2014/1/14	82	108	68	2.4	74	24	轻度污染

预备知识

数据集划分方法

- **基本准则：保持训练集、验证集、测试集之间的互斥性。**
- **参考原则：**

1、对于小规模样本集（几万量级），常用的分配比例是 60% 训练集、20% 验证集、20% 测试集。

2、对于大规模样本集（百万级以上），只要验证集和测试集的数量足够即可，例如有 100w 条数据，那么留 1w 验证集，1w 测试集即可。1000w 的数据，同样留 1w 验证集和 1w 测试集。

3、超参数越少，或者超参数很容易调整，那么可以减少验证集的比例，更多的分配给训练集。

预备知识

数据集划分方法

◆ **划分方法：** **K折交叉验证**（一种动态验证的方式，这种方式可以降低数据划分带来的影响）

以10折交叉验证为例，具体步骤如下：

1、将数据集分为训练集和测试集，**将测试集放在一边**；

2、将训练集平均分成不相交的10个子集；

3、每一次挑选其中的1份作为验证集，其余的9份作为训练集进行模型训练，得到模型以及评价指标；

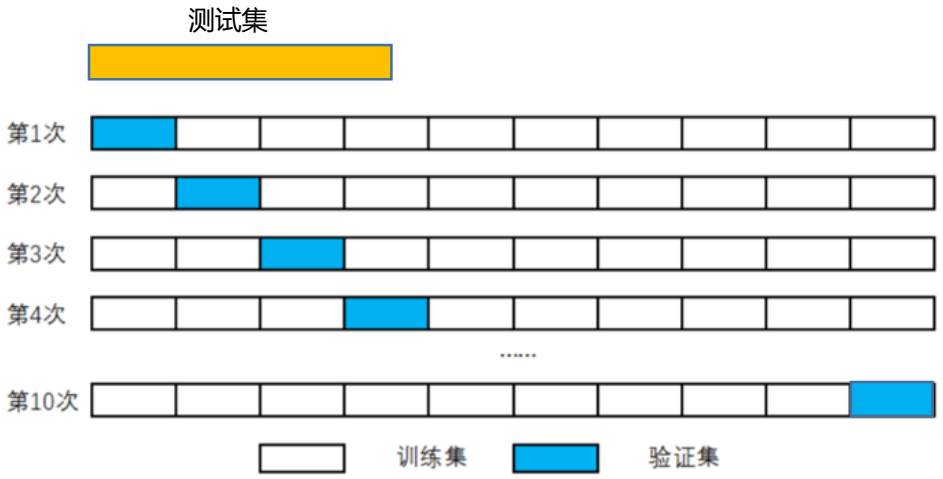
4、重复第3步10次，通过 10 次训练后，得到了 10个不同的模型；

5、将10个模型的评价指标取平均值，作为交叉验证的评估指标；

6、使用不同的超参数，重复以上2-5步，根据最好的交叉验证评估指标，挑选出最优的超参数;

7、使用最优的超参数，将数据全部作为训练集重新训练模型；

8、最后使用测试集测试**评估模型**，计算**评价指标。**



预备知识

数据集划分方法

◆ **实现方法：**比如 sklearn中的model_selection.KFold函数，格式如下：

```
sklearn.model_selection.KFold(n_splits=3, shuffle=False, random_state=None)
```

参数说明	含义
n_splits	分为几折交叉验证
shuffle	在每次划分时，是否进行洗牌。 若为Falses时，其效果等同于random_state等于整数，每次划分的结果相同； 若为True时，每次划分的结果都不一样，表示经过洗牌，随机取样的
random_state	随机种子数（设置了这个参数之后，每次生成的结果是一样的，而且设置了random_state之后就没必要设置shuffle）

代码示例：

```
# 导入包
from sklearn.model_selection import KFold
import numpy as np
# 构建数据集
X = np.arange(24).reshape(12,2)
Y = np.arange(12).reshape(12,1)

#调用k折交叉验证方法
kf = KFold(n_splits=3,shuffle=False)
for train_index,valid_index in kf.split(X):
    print("TRAIN:", train_index, "VALID:", valid_index)
  X_train, X_valid = X[train_index], X[valid_index]
  Y_train, Y_valid = Y[train_index], Y[valid_index]
```

运行结果：

```
TRAIN: [ 4  5  6  7  8  9 10 11] VALID: [0 1 2 3]
TRAIN: [ 0  1  2  3  8  9 10 11] VALID: [4 5 6 7]
TRAIN: [0 1 2 3 4 5 6 7] VALID: [ 8  9 10 11]
```

预备知识

评分模型

对于**多分类**任务中，常用的评价指标有**宏平均**（Macro-Averaging）、**微平均**（Micro-Averaging）、**加权平均。**

- 宏平均（Macro-Averaging）是指所有类别的每一个统计指标值的算数平均值，也就是宏精确率（Macro-Precision），宏召回率（Macro-Recall），宏F值（Macro-F Score）。
- 微平均（Micro-Averaging）是对数据集中的每一个示例不分类别进行统计建立全局混淆矩阵，然后计算相应的指标。

$$P_{macro} = \frac{1}{n}\sum_{i=1}^{n} P_i$$

$$P_{micro} = \frac{\bar{TP}}{\bar{TP} + \bar{FP}} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}$$

$$R_{macro} = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$R_{micro} = \frac{\bar{TP}}{\bar{TP} + \bar{FN}} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i}$$

$$F_{macro} = \frac{2 \times P_{macro} \times R_{macro}}{P_{macro} + R_{macro}}$$

$$F_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}}$$

预备知识

评分模型

对于多分类任务中，常用的评价指标有**宏平均**（Macro-Averaging）、**微平均**（Micro-Averaging）、**加权平均。**

- 加权平均：是指所有类别的每一个统计指标值按照各自类别占测试集的比例，做加权计算，得到加权精确率，加权召回率，加权F值。比如 sklearn中的**metrics库**有两种方法计算**加权F值**，格式如下：

```
metrics.f1_score(y_true, y_pred, average='weighted')
```

```
metrics.classification_report(y_true, y_pred, labels=None, target_names=None, sample_weight=None, digits=2)
```


```
加权F值： 0.8937367776107534
```

	precision	recall	f1-score	support
中度污染	0.84	0.76	0.80	34
优	0.90	0.96	0.93	103
良	0.90	0.94	0.92	189
轻度污染	0.90	0.81	0.85	95
重度污染	1.00	0.67	0.80	9
accuracy			0.90	430
macro avg	0.91	0.83	0.86	430
weighted avg	0.90	0.90	0.89	430

实验步骤

◆ 实验步骤 （使用Python自编程）

1、准备数据

✓读取数据，清理记录为0的数据，并提取合适有用的特征；

✓将数据分割为**训练集、验证集、测试集**

2、定义模型

k-近邻算法的具体步骤如下：

1）计算待分类样本点与所有已标注样本点之间的距离

2）按照距离从小到大排序

3）选取与待分类样本点距离最小的k个点

4）确定前k个点中，每个类别的出现次数

5）返回次数最高的那个类别

3、训练模型

✓使用训练集训练模型，调整k值，找到合适模型，使用验证集验证模型

4、评估模型

✓自定义评价指标，使用测试集评估模型

实验要求

1、使用K折交叉验证方法，划分训练集和验证集（可调库）；

2、记录调参过程和结果，根据评价指标，选出最合适的K值；

3、使用加权平均指标来评价模型（可调库）；

4、使用测试集评估模型，要求加权F值指标>0.85。

注意事项

◆1、数据集中的0记录要清理掉

	A	B	C	D	E	F	G	H
1	日期	PM2.5	PM10	SO2	CO	NO2	O3	质量等级
674	2015/11/4	210	0	15	2	108	29	重度污染
675	2015/11/5	110	0	6	1	53	26	轻度污染
676	2015/11/6	20	9	2	0.5	29	38	优
677	2015/11/7	19	0	2	0.5	29	41	优
678	2015/11/8	60	89	3	0.9	46	37	良
679	2015/11/9	124	0	4	1.6	56	7	中度污染
680	2015/11/10	132	0	4	1.6	55	5	中度污染
681	2015/11/11	104	0	4	1.9	55	11	轻度污染
682	2015/11/12	155	136	6	2.5	60	3	重度污染
683	2015/11/13	208	188	15	3.1	70	4	重度污染
684	2015/11/14	274	298	17	3.6	83	11	严重污染
685	2015/11/15	196	0	13	3.2	70	31	重度污染

◆2、Python编程的warning日志，可以加如下图代码忽略掉

```
import warnings
warnings.filterwarnings(action = 'ignore')
```

◆3、绘图时显示中文乱码，可以加两行代码解决

```
plt.rcParams['font.sans-serif']=['SimHei']   #解决中文显示乱码问题
plt.rcParams['axes.unicode_minus']=False
```

提交方式

实验报告提交至平台 http://grader.tery.top:8000/#/courses

注意：

- 1、用户名、密码默认均为学号（若之前有修改过密码的，请用新密码登陆）；
- 2、请提交到相应的条目「2022春统计机器学习」课程 - 实验三；
- 3、提交截止时间：下周四 晚24点前；
- 4、文件夹&压缩包命名要求： 学号_姓名_统计机器学习实验三
- 5、提交内容：实验报告(.pdf文件)+代码(.py文件)，一起打包为zip格式压缩包。

统计机器学习实验

同学们，请开始实验吧！