



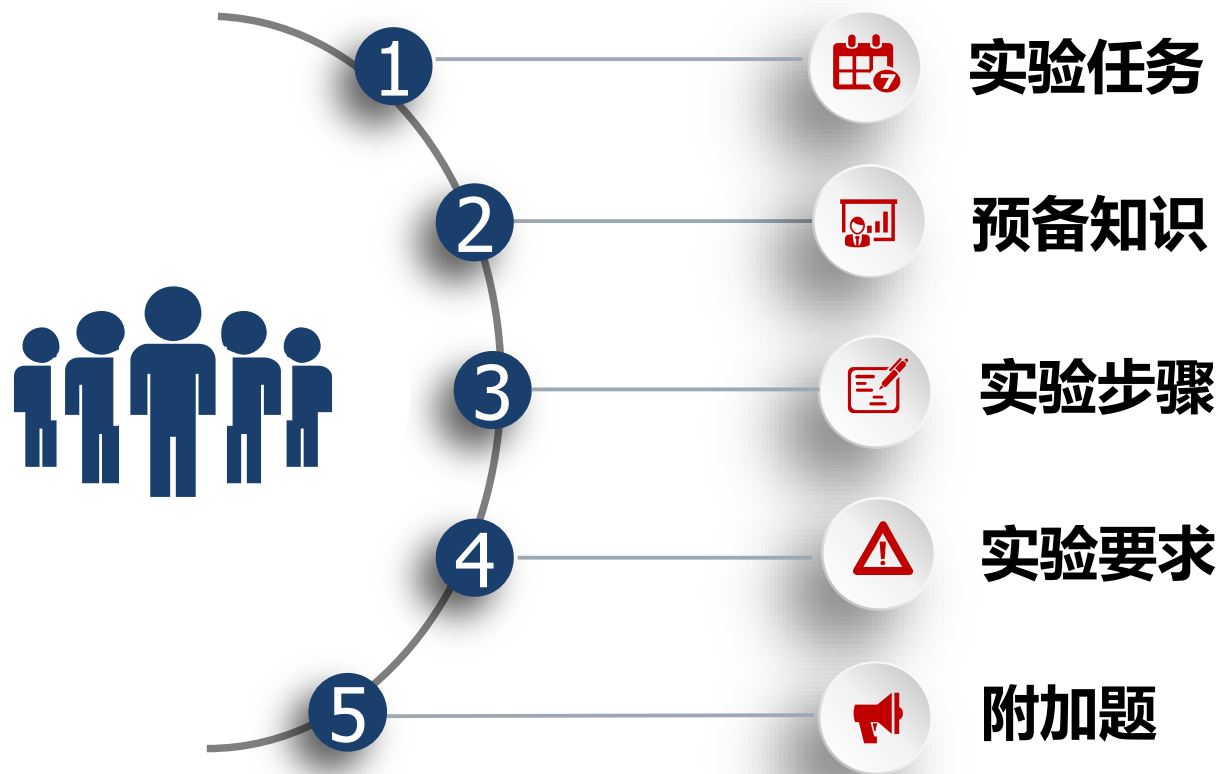
哈爾濱工業大學(深圳)  
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

# 实验四：使用K-近邻模型实现 空气质量的预测

《统计机器学习》

2024年春

# 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 30 分。

实验项目	一	二	三	四	五
学时	2	2	2	2	2
实验内容	Python基础实践	感知机模型	决策树模型	K近邻模型	支持向量机模型
分数	4	6	7	6	7
上课时间	第11周	第13周	第14周	第15周	第16周
检查方式	提交实验截图文档		提交实验报告、工程文件		

# 一、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。
- ◆ **任务二：**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

## 二、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：

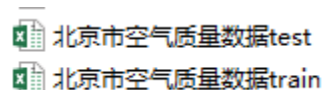
日期、PM2.5、PM10、SO2、CO、NO2、O3

- 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

# 三、预备知识4

## 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。

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

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

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

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

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

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



# 三、预备知识5



## 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

# 四、实验步骤

## ◆ 实验步骤（使用Python自编程）

### 1、准备数据

- ✓ 读取数据，处理记录为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（可调库），使用测试集评估模型



# 五、实验要求

除了实验三介绍的sklearn中的model\_selection.Kfold方法外，还需要调研其他方法来实现交叉验证思想，总结分析对比。

- 1、使用交叉验证思想，划分训练集和验证集；
- 2、使用加权平均指标来评价模型（可调库）
- 3、记录调参过程和结果，根据评价指标，选出最合适的K值；
- 4、使用测试集评估模型，要求加权F值指标 $>0.85$ 。



## 六、注意事项

- ◆ 1、数据集中的0记录**处理**（比如替换/插值/删除等等，可参考附录中的方法）

	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://labgrader.hitsz.edu.cn: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「2024春-统计机器学习-数学1&2」课程 - 实验四；
- 3、提交截止时间：下周二 6月18号 晚24点；
- 4、文件夹&压缩包命名要求：学号\_姓名\_统计机器学习实验四
- 5、提交内容：实验报告(.pdf文件)+代码(.py/ipynb文件)，一起打包为zip格式压缩包。

其他：

- 1) 数学1&2班 作业提交至课程「2024春-统计机器学习-数学1&2」
- 2) 数学3&4班 作业提交至课程「2024春-统计机器学习-数学3&4」
- 3) 计算机/通信/机械/自动化/光电/电气等专业 作业提交至课程「2024春-统计机器学习-综合班」
- 4) 每位同学都只会显示一个统计机器学习课程的，对上实验几提交即可。

# 附录——缺失值处理

- 数据中的某个或某些特征的值是不完整的，这些值称为缺失值。对缺失值处理前需先识别缺失值，pandas提供isnull方法，能够识别出缺失值，返回bool。isnull方法结合其他操作，找出缺失值的数量及占比，如下代码所示。

```
In[1]: import pandas as pd
dit = {'col1': [0, 1, 2, None, 4], 'col2': [5, None, 6, 7, None]}
df = pd.DataFrame(dit)
print('缺失值数量为: \n', df.isnull().sum())
```

```
Out[1]: 缺失值数量为:
col1    1
col2    2
dtype: int64
```

```
In[2]: print('缺失值占比为: \n', df.isnull().sum() / len(df))
```

```
Out[2]: 缺失值占比为:
col1    0.2
col2    0.4
dtype: float64
```

# 附录——缺失值处理

## 1. 删除法

- 删除法是指将含有缺失值的特征或者记录删除。删除法分为删除观测记录和删除特征两种，观测记录指删除行，特征指删除列，它属于利用减少样本量来换取信息完整度的一种方法，是一种最简单的缺失值处理方法。pandas提供简便的删除缺失值的方法**dropna**，通过参数控制，该方法既可以删除观测记录，亦可以删除特征，其基本语法格式如下。

```
pandas.DataFrame.dropna(axis=0,      how='any',      thresh=None,      subset=None,
inplace=False)
```

- **dropna**方法常用的参数及其说明，如下表示。

参数名称	说明
axis	接收0或1。表示轴向，0为删除观测记录（行），1为删除特征（列）。默认为0
how	接收特定str。表示删除的形式。any表示只要有缺失值存在就执行删除操作。all表示当且仅当全部为缺失值时执行删除操作。默认为any
subset	接收类array数据。表示进行删除缺失值的列。默认为None，表示所有列
inplace	接收bool。表示是否在原表上进行操作。默认为False

# 附录——缺失值处理

## 2. 替换法

- 替换法是指用一个特定的值替换缺失值。特征可分为数值型和类别型，两者出现缺失值时的处理方法也是不同的。缺失值所在特征为数值型时，通常利用其均值、中位数和众数等描述其集中趋势的统计量来代替缺失值；缺失值所在特征为类别型时，则选择使用众数来替换缺失值。fillna方法用于替换缺失值，其基本语法格式如下。

```
DataFrame.fillna(value=None, method=None, axis=None, inplace=False, limit=None)
```

- **fillna**方法常用的参数及其说明，如下表所示。

参数名称	说明
value	接收数字，dict，Series或者DataFrame。表示用来替换缺失值的值。无默认值
method	接收特定str。表示确实值填充的方法，当value参数未填写时起效。取值为“backfill”或“bfill”时，表示使用下一个非缺失值填补缺失值；取值为“pad”或“ffill”时，表示使用上一个非缺失值填补缺失值。默认为None
axis	接收0或1。表示轴向。默认为1
inplace	接收bool。表示是否在原表上进行操作。默认为False
limit	接收int。表示填补缺失值个数上限，超过则不进行填补。默认为None

# 附录——缺失值处理

## 3. 插值法

- 删除法简单易行，但是会引起数据结构变动，样本减少；替换法使用难度较低，但是会影响数据的标准差，导致信息量变动。在面对数据缺失问题时，除了这两种方法之外，还有一种常用的方法——插值法。
- interpolate方法用于对缺失值进行插值。针对DataFrame的interpolate方法，其基本语法格式如下。

```
DataFrame.interpolate(method='linear', axis=0, limit=None, inplace=False, limit_direction='forward', limit_area=None, downcast=None, **kwargs)
```

- **interpolate**方法常用的参数及其说明，如下表所示。

参数名称	说明
method	接收str。表示插值方法。默认为“linear”
axis	接收int。表示插值的轴。默认为0
limit	接收int。表示遇到连续NaN插值的最大数。默认为None
inplace	接收bool。表示是否更新原DataFrame。默认为False

# 附录——缺失值处理

## 3. 插值法

➤ interpolate方法提供了多种插值方法，常用插值方法及使用场景，如下表所示。

方法	说明
linear	线性插值。忽视索引，将所有值看做等距隔开。若DataFrame或Series为多重索引，则只支持此种插值方法
time	时间插值。索引为时间类型，按给定时间间隔插值
Index、values	索引插值。按照数值化的索引值来插值



# 统计机器学习实验

---

同学们，请开始实验吧！