实验一：Python基础实践

《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、实验目标

- 掌握Windows系统下Anaconda的安装
- 掌握Jupyter Notebook的使用方法
- 熟悉Python的基础知识
- 掌握Numpy中数组矩阵的运算
- 掌握Pandas中DataFrame的常用索引方法
- 掌握Matplotlib中常见图形的作用与绘制方法
- 熟悉sklearn库内分类、回归、聚类方法

二、Python基础

- （1）Python：机器学习的首选工具
- （2） Python的集成开发环境：Anaconda
- （3） Python第三方包的引用
- （4） Numpy库介绍和使用示例
- （5） Pandas库介绍和使用示例
- （6） Numpy和Pandas的综合应用
- （7） Matplotlib的综合应用
- （8） Sklearn库相关介绍

二、Python基础

Python：机器学习的首选工具

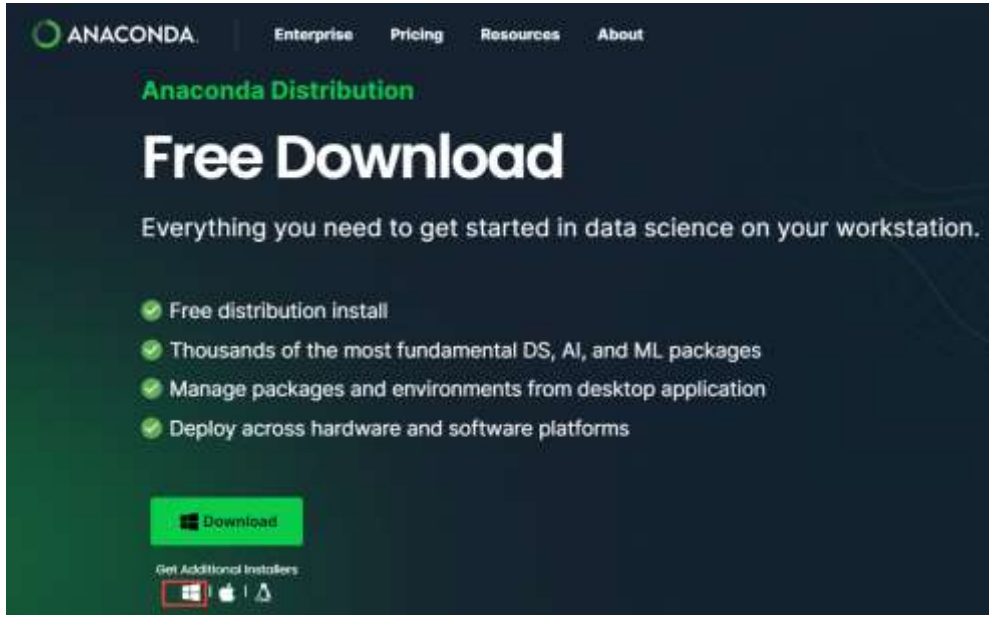
- **Python是一款面向对象的解释型计算机语言。开源、代码可读性强，可实现高效开发等是Python的重要特征**
  - 面向对象的程序设计（Object Oriented Programming，OOP）是相对面向过程的程序设计而言的。
  - OOP采用“封装”的思想，将具有一定独立性和通用性的处理过程和变 量（数据），封装在“对象”中
  - 变量称为对象的“**属性**”，变量值对应属性值（有具体变量值的对象称为“对象实 例”）；处理过程称为对象的“**方法**”；多个具有内在联系的对象可进一步封装在“**类**”中。
- **Python在机器学习领域获得广泛使用的原因：**
  - 简明易用，严谨专业；
  - 良好的开发社区生态；
  - 丰富的第三方程序包
    - 尤其是Numpy、Pandas、Scipy、Matplotlib、Scikit-learn等第三方包在数据组织、 科学计算、可视化、机器学习等方面有着极为成熟、丰富和卓越的表现。
    - 在NumPy和SciPy基础上开发的Scikit-learn，是专门面向机器学习的Python第三方包。

二、Python基础

Python的集成开发环境：Anaconda

- Anaconda简介：是一款兼容Linux、Windows 和Mac OS 环境，支持Python2系列和Python3系列的，且可方便快捷完成机器学习和数据科学任务的开源IDE。**Anaconda的官方下载地址为：**

  https://www.anaconda.com/download

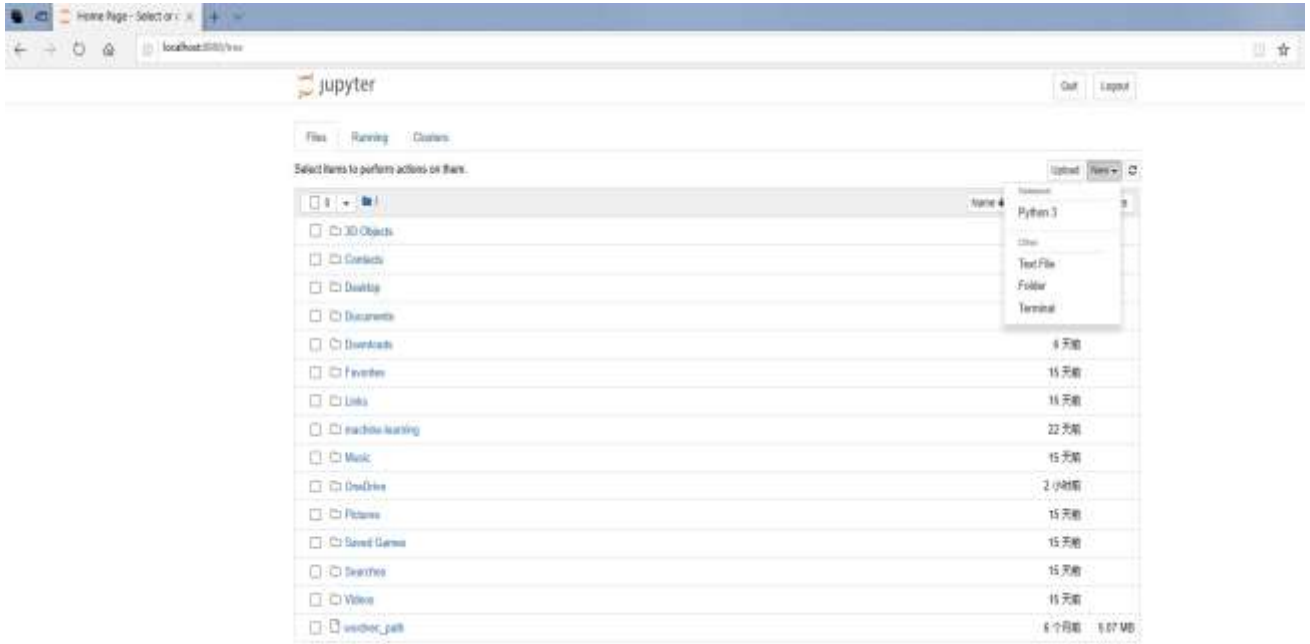


二、Python基础

Python的集成开发环境：Anaconda

◆ Jupyter Notebook的使用

- **是一个基于网页的交互式笔记本；扩展名为.ipynb格式文件**



安装 Anaconda（包含python环境+Jupyter编辑器）

等同

单独安装 Python 环境 +其他 IDE(Pycharm / VScode)

- **使用教程** Anaconda3内jupyter notebook使用教程

二、Python基础

Python第三方包的引用

- 第三方包以模块（Moduel，文件扩展名为.py）方式，将可实现各种功能的 程序代码（变量、函数）“打包”在一起。
- 引用第三方包中的模块的基本函数是：import函数。
  - 之后可在自己编写的Python程序直接调用已导入模块中的函数，通过代 码重用（重复使用）的方式快速实现某种特定功能。
  - 模块导入的一般语法格式为：
    - import 模块名：导入指定模块
    - from 模块名 import 函数名：导入指定模块中的指定函数
    - from 模块名 import 函数名1,函数名2,...：导入指定模块中的若干个指定函数
    - from 模型名 import *：导入指定模块中的所有函数
    - 可增加：as 别名。例如：import numpy as np，表示导入numpy并指定别名为np。 指定别名可以有效避免不同模块有相同函数名的问题。

二、Python基础

Numpy库介绍

- Numpy（Numerical Python的简称）是高性能科学计算和数据分析的基础包;
- Numpy 最重要的一个特点是其 N 维数组对象 **ndarray**，它是一系列同类型数据的集合，以 0 下标为开始进行集合中元素的索引;
- ndarray对象的内容可以通过**索引或切片**来访问和修改，与 Python 中 list 的切片操作一样。

- 先在Windows系统终端内执行安装命令 pip install numpy
- 然后在python代码中导入numpy库 import numpy as np

```
C:\Users\lenovo>conda list
# packages in environment at E:\anaconda3:
#
# Name                    Version                   Build  Channel
_ipyw_jlab_nb_ext_conf    0.1.0                    py37_0
alabaster                 0.7.12                   py37_0
anaconda                  2020.02                  py37_0
anaconda-client           1.7.2                    py37_0
anaconda-navigator        1.9.12                   py37_0
anaconda-project          0.8.4                      py_0
argh                      0.26.2                   py37_0
asn1crypto                1.3.0                    py37_0
astroid                   2.3.3                    py37_0
astropy                   4.0                 py37he774522_0
atomicwrites              1.3.0                    py37_1
attrs                     19.3.0                     py_0
```

注意：**Anaconda**环境中默认是有常见的第三方库，在终端上可通过 **conda list** 命令查看，不用自己安装**numpy**库

二、Python基础

Numpy库的基本操作

- 查看数组属性
- np.size、np.shape、np.ndim、 np.dtype

- 索引
- b= np.arange(5,20,2)、b[5]、b[2:5]

- 切片
- arr_slice = b[2:5]、 arr_slice[1] = 12345、 arr_slice[:]=64

更多学习请参考numpy中文网：https://www.numpy.org.cn/

二、Python基础

Numpy使用示例

- 创建和访问Numpy的一维数组和二维数组

```
import numpy as np
data=np.array([1,2,3,4,5,6,7,8,9])
print('Numpy的1维数组:\n{0}'.format(data))
print('数据类型:%s'%data.dtype)
print('1维数组中各元素扩大10倍:\n{0}'.format(data*10))
print('访问第2个元素：{0}'.format(data[1]))
data=np.array([[1,3,5,7,9],[2,4,6,8,10]])
print('Numpy的2维数组:\n{0}'.format(data))
print('访问2维数组中第1行第2列元素：{0}'.format(data[0,1]))
print('访问2维数组中第1行第2至5列元素：{0}'.format(data[0,1:4]))
print('访问2维数组中第1行上的所有元素：{0}'.format(data[0,:]))
```

```
Numpy的1维数组:
[1 2 3 4 5 6 7 8 9]
数据类型:int32
1维数组中各元素扩大10倍:
[10 20 30 40 50 60 70 80 90]
访问第2个元素：2
Numpy的2维数组:
[[ 1  3  5  7  9]
 [ 2  4  6  8 10]]
访问2维数组中第1行第2列元素：3
访问2维数组中第1行第2至5列元素：[3 5 7]
访问2维数组中第1行上的所有元素：[1 3 5 7 9]
```

- Numpy数组的数据来源：Python列表

```
data=[[1,2,3,4,5,6,7,8,9],['A','B','C','D','E','F','G','H','I']]
print('data是Python的列表(list):\n{0}'.format(data))
MyArray1=np.array(data)
print('MyArray1是Numpy的N维数组:\n%s\nMyarray1的形状:%s'%(MyArray1,MyArray1.shape))
```

```
data是Python的列表(list):
[[1, 2, 3, 4, 5, 6, 7, 8, 9], ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I']]
MyArray1是Numpy的N维数组:
[['1' '2' '3' '4' '5' '6' '7' '8' '9']
 ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I']]
Myarray1的形状:(2, 9)
```

代码详细注释请查看老师下发的**.ipynb**文件。

lab1-numpy使用示例.ipynb

二、Python基础

Numpy使用示例——Numpy的计算功能

- 数组计算

```
MyArray2=np.arange(10)
print('MyArray2:\n{0}'.format(MyArray2))
print('MyArray2的基本描述统计量:\n均值：%f，标准差：%f，总和：%f，最大值：%f'%(MyArray2.mean(),MyArray2.std(),MyArray2.sum(),MyArray2.max
print('MyArray2的累计和：{0}'.format(MyArray2.cumsum()))
print('MyArray2开平方:{0}'.format(np.sqrt(MyArray2)))
np.random.seed(123)
MyArray3=np.random.randn(10)
print('MyArray3:\n{0}'.format(MyArray3))
print('MyArray3排序结果:\n{0}'.format(np.sort(MyArray3)))
print('MyArray3四舍五入到最近整数:\n{0}'.format(np.rint(MyArray3)))
print('MyArray3各元素的正负号:{0}'.format(np.sign(MyArray3)))
print('MyArray3各元素非负数的显示"正"，负数显示"负":\n{0}'.format(np.where(MyArray3>0,'正','负')))
print('MyArray2+MyArray3的结果:\n{0}'.format(MyArray2+MyArray3))
```

```
MyArray2:
[0 1 2 3 4 5 6 7 8 9]
MyArray2的基本描述统计量:
均值：4.500000，标准差：2.872281，总和：45.000000，最大值：9.000000
MyArray2的累计和：[ 0  1  3  6 10 15 21 28 36 45]
MyArray2开平方:[0.         1.         1.41421356 1.73205081 2.         2.23606798
 2.44948974 2.64575131 2.82842712 3.        ]
MyArray3:
[-1.0856306   0.99734545  0.2829785  -1.50629471 -0.57860025  1.65143654
 -2.42667924 -0.42891263  1.26593626 -0.8667404 ]
MyArray3排序结果:
[-2.42667924 -1.50629471 -1.0856306  -0.8667404  -0.57860025 -0.42891263
  0.2829785   0.99734545  1.26593626  1.65143654]
MyArray3四舍五入到最近整数:
[-1.  1.  0. -2. -1.  2. -2. -0.  1. -1.]
MyArray3各元素的正负号:[-1.  1.  1. -1. -1.  1. -1. -1.  1. -1.]
MyArray3各元素非负数的显示"正"，负数显示"负":
['负' '正' '正' '负' '负' '正' '负' '负' '正' '负']
MyArray2+MyArray3的结果:
[-1.0856306   1.99734545  2.2829785   1.49370529  3.42139975  6.65143654
  3.57332076  6.57108737  9.26593626  8.1332596 ]
```

- 创建矩阵和矩阵乘法

```
np.random.seed(123)
X=np.floor(np.random.normal(5,1,(2,5)))
Y=np.eye(5)
print('X:\n{0}'.format(X))
print('Y:\n{0}'.format(Y))
print('X和Y的点积：\n{0}'.format(np.dot(X,Y)))
```

```
X:
[[3. 5. 5. 3. 4.]
 [6. 2. 4. 6. 4.]]
Y:
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
X和Y的点积：
[[3. 5. 5. 3. 4.]
 [6. 2. 4. 6. 4.]]
```

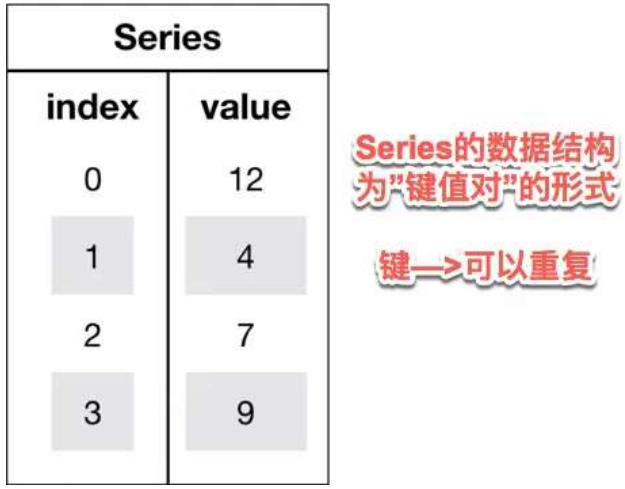
- 矩阵运算初步

```
from numpy.linalg import inv,svd,eig,det
X=np.random.randn(5,5)
print(X)
mat=X.T.dot(X)
print(mat)
print('矩阵mat的逆：\n{0}'.format(inv(mat)))
print('矩阵mat的行列式值：\n{0}'.format(det(mat)))
print('矩阵mat的特征值和特征向量：\n{0}'.format(eig(mat)))
print('对矩阵mat做奇异值分解：\n{0}'.format(svd(mat)))
```

二、Python基础

Pandas库介绍

- Pandas是Python第三方库，提供高性能易用数据类型和分析工具
- Pandas中有两大核心数据结构：Series（一维数据） 和DataFrame（多特征数据，既有行索引，又有列索引）

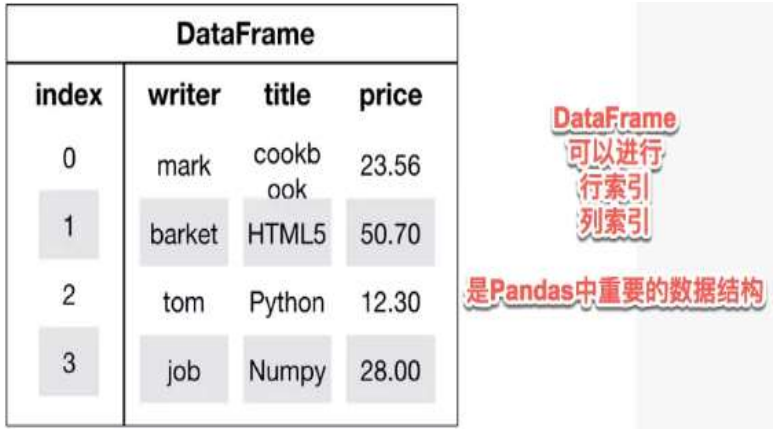


Series	
index	value
0	12
1	4
2	7
3	9

Series的数据结构为"键值对"的形式

键—>可以重复

- 安装命令 pip install pandas （使用Anaconda自带这个库，就不用安装啦）
- 导入pandas库 import pandas as pd



DataFrame			
index	writer	title	price
0	mark	cookbook	23.56
1	barket	HTML5	50.70
2	tom	Python	12.30
3	job	Numpy	28.00

DataFrame可以进行行索引列索引

是Pandas中重要的数据结构

二、Python基础

Pandas库的基本操作

- 查看数据
  df.describe()、df.index 、df.columns 、df.values
- 选取特定列和行的数据
  df[col]、df[[col1,col2]]、df.iloc[[row1, row2], [col1, col2]]
- 增加、删除、修改列的值
  df.append( ) 、del pd[col] 、df.pop
- 导入导出文件
  df.read_csv、df.to_csv

更多学习请参考pandas中文网： https://www.pypandas.cn/

二、Python基础

Pandas使用示例

- Pandas的序列和索引

```
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
data=Series([1,2,3,4,5,6,7,8,9],index=['ID1','ID2','ID3','ID4','ID5','ID6','ID7','ID8','ID9'])
print('序列中的值:\n{0}'.format(data.values))
print('序列中的索引:\n{0}'.format(data.index))
print('访问序列的第1和第3上的值:\n{0}'.format(data[[0,2]]))
print('访问序列索引为ID1和ID3上的值:\n{0}'.format(data[['ID1','ID3']]))
print('判断ID索引是否存在:%s；判断ID10索引是否存在:%s'%('ID1' in data,'ID10' in data))
```

```
序列中的值:
[1 2 3 4 5 6 7 8 9]
序列中的索引:
Index(['ID1', 'ID2', 'ID3', 'ID4', 'ID5', 'ID6', 'ID7', 'ID8', 'ID9'], dtype='object')
访问序列的第1和第3上的值:
ID1    1
ID3    3
dtype: int64
访问序列索引为ID1和ID3上的值:
ID1    1
ID3    3
dtype: int64
判断ID索引是否存在:True；判断ID10索引是否存在:False
```

- Pandas的数据框

```
import pandas as pd
from pandas import Series,DataFrame
data=pd.read_excel('北京市空气质量数据.xlsx')
print('date的类型：{0}'.format(type(data)))
print('数据框的行索引：{0}'.format(data.index))
print('数据框的列名：{0}'.format(data.columns))
print('访问AQI和PM2.5所有值：\n{0}'.format(data[['AQI','PM2.5']]))
print('访问第2至3行的AQI和PM2.5：\n{0}'.format(data.loc[1:2,['AQI','PM2.5']]))
print('访问索引1至索引2的第2和4列：\n{0}'.format(data.iloc[1:3,[1,3]]))
data.info()
```

```
date的类型：<class 'pandas.core.frame.DataFrame'>
数据框的行索引：RangeIndex(start=0, stop=2155, step=1)
数据框的列名：Index(['日期', 'AQI', '质量等级', 'PM2.5', 'PM10', 'SO2', 'CO', 'NO2', 'O3'], dtype='object')
访问AQI和PM2.5所有值：
    AQI  PM2.5
0    81     45
1   145    111
2    74     47
3   149    114
4   119     91
```

- Pandas的数据加工处理

```
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
df1=DataFrame({'key':['a','d','c','a','b','d','c'],'var1':range(7)})
df2=DataFrame({'key':['a','b','c','c'],'var2':[0,1,2,2]})
df=pd.merge(df1,df2,on='key',how='outer')
df.iloc[0,2]=np.NaN
df.iloc[5,1]=np.NaN
print('合并后的数据:\n{0}'.format(df))
df=df.drop_duplicates()
print('删除重复数据行后的数据:\n{0}'.format(df))
print('判断是否为缺失值:\n{0}'.format(df.isnull()))
print('判断是否不为缺失值:\n{0}'.format(df.notnull()))
print('删除缺失值后的数据:\n{0}'.format(df.dropna()))
fill_value=df[['var1','var2']].apply(lambda x:x.mean())
print('以均值替换缺失值:\n{0}'.format(df.fillna(fill_value)))
```

合并后的数据：

	key	var1	var2
0	a	0.0	NaN
1	a	3.0	0.0
2	d	1.0	NaN
3	d	5.0	NaN
4	c	2.0	2.0
5	c	NaN	2.0
6	c	6.0	2.0
7	c	6.0	2.0
8	b	4.0	1.0

删除重复数据行后的数据：

	key	var1	var2
0	a	0.0	NaN
1	a	3.0	0.0
2	d	1.0	NaN
3	d	5.0	NaN
4	c	2.0	2.0
5	c	NaN	2.0
6	c	6.0	2.0
8	b	4.0	1.0

判断是否为缺失值：

	key	var1	var2
0	False	False	True
1	False	False	False
2	False	False	True
3	False	False	True
4	False	False	False
5	False	True	False
6	False	False	False
8	False	False	False

判断是否不为缺失值：

	key	var1	var2
0	True	True	False
1	True	True	True
2	True	True	False
3	True	True	False
4	True	True	True
5	True	False	True
6	True	True	True
8	True	True	True

删除缺失值后的数据：

	key	var1	var2
1	a	3.0	0.0
4	c	2.0	2.0
6	c	6.0	2.0
8	b	4.0	1.0

以均值替换缺失值：

	key	var1	var2
0	a	0.0	1.4
1	a	3.0	0.0
2	d	1.0	1.4
3	d	5.0	1.4
4	c	2.0	2.0
5	c	3.0	2.0
6	c	6.0	2.0
8	b	4.0	1.0

代码详细注释请查看老师下发的**.ipynb**文件。

lab1-pandas使用示例.ipynb

二、Python基础

Numpy和Pandas的综合应用——空气质量监测数据的预处理和基本分析

- 数据预处理

```python
import numpy as np
import pandas as pd
from pandas import Series,DataFrame

data=pd.read_excel('北京市空气质量数据.xlsx')
data=data.replace(0,np.NaN)
data['年']=data['日期'].apply(lambda x:x.year)
month=data['日期'].apply(lambda x:x.month)
quarter_month={'1':'一季度','2':'一季度','3':'一季度',
               '4':'二季度','5':'二季度','6':'二季度',
               '7':'三季度','8':'三季度','9':'三季度',
              '10':'四季度','11':'四季度','12':'四季度'}
data['季度']=month.map(lambda x:quarter_month[str(x)])
bins=[0,50,100,150,200,300,1000]
data['等级']=pd.cut(data['AQI'],bins,labels=['一级优','二级良','三级轻度污染','四级中度污染','五级重度污染','六级严重污染'])
print('对AQI的分组结果：\n{0}'.format(data[['日期','AQI','等级','季度']]))
```

对AQI的分组结果：

	日期	AQI	等级	季度
0	2014-01-01	81.0	二级良	一季度
1	2014-01-02	145.0	三级轻度污染	一季度
2	2014-01-03	74.0	二级良	一季度
3	2014-01-04	149.0	三级轻度污染	一季度
4	2014-01-05	119.0	三级轻度污染	一季度
...	...	...	...	...
2150	2019-11-22	183.0	四级中度污染	四季度
2151	2019-11-23	175.0	四级中度污染	四季度
2152	2019-11-24	30.0	一级优	四季度
2153	2019-11-25	40.0	一级优	四季度
2154	2019-11-26	73.0	二级良	四季度

- 基本分析

```python
print('各季度AQI和PM2.5的均值:\n{0}'.format(data.loc[:,['AQI','PM2.5']].groupby(data['季度']).mean()))
print('各季度AQI和PM2.5的描述统计量:\n',data.groupby(data['季度'])['AQI','PM2.5'].apply(lambda x:x.describe()))

def top(df,n=10,column='AQI'):
    return df.sort_values(by=column,ascending=False)[:n]
print('空气质量最差的5天:\n',top(data,n=5)[['日期','AQI','PM2.5','等级']])
print('各季度空气质量最差的3天:\n',data.groupby(data['季度']).apply(lambda x:top(x,n=3)[['日期','AQI','PM2.5','等级']]))
print('各季度空气质量情况:\n',pd.crosstab(data['等级'],data['季度'],margins=True,margins_name='总计',normalize=False))
```

各季度AQI和PM2.5的均值:

季度	AQI	PM2.5
一季度	109.327778	77.225926
三季度	98.911071	49.528131
二季度	109.389004	55.149723
四季度	109.612403	77.195736

各季度AQI和PM2.5的描述统计量:

季度		AQI	PM2.5
一季度	count	540.000000	540.000000
	mean	109.327778	77.225926
	std	80.495408	73.133857
	min	26.000000	4.000000
	25%	65.000000	24.000000
	50%	80.000000	53.000000
	75%	145.000000	109.250000
	max	470.000000	454.000000
三季度	count	551.000000	551.000000
	mean	98.911071	49.528131
	std	65.494518	35.394897
	min	28.000000	3.000000
	25%	60.000000	23.000000
	50%	93.000000	41.000000
	75%	130.500000	67.000000
	max	252.000000	202.000000
二季度	count	542.000000	541.000000
	mean	109.389004	55.149723
	std	69.809042	35.918345

空气质量最差的5天:

	日期	AQI	PM2.5	等级
1218	2017-05-04	500.0	NaN	六级严重污染
723	2015-12-23	483.0	477.0	六级严重污染
699	2015-12-01	476.0	464.0	六级严重污染
1095	2017-01-01	470.0	454.0	六级严重污染
696	2015-11-30	450.0	343.0	六级严重污染

各季度空气质量最差的3天:

季度		日期	AQI	PM2.5	等级
一季度	1095	2017-01-01	470.0	454.0	六级严重污染
	45	2014-02-15	426.0	380.0	六级严重污染
	55	2014-02-25	403.0	354.0	六级严重污染
三季度	198	2014-07-06	252.0	202.0	五级重度污染
	211	2014-07-21	245.0	195.0	五级重度污染
	183	2014-07-03	240.0	190.0	五级重度污染
二季度	1218	2017-05-04	500.0	NaN	六级严重污染
	1219	2017-05-05	342.0	181.0	六级严重污染
	103	2014-04-14	278.0	229.0	五级重度污染
四季度	723	2015-12-23	483.0	477.0	六级严重污染
	699	2015-12-01	476.0	464.0	六级严重污染
	696	2015-11-30	450.0	343.0	六级严重污染

各季度空气质量情况:

季度	一季度	三季度	二季度	四季度	总计
等级					
一级优	145	96	38	108	387
二级良	170	209	240	230	849
三级轻度污染	99	184	132	64	479
四级中度污染	57	72	96	33	258
五级重度污染	48	10	14	58	130
六级严重污染	21	0	2	23	46
总计	540	551	542	550	2183

二、Python基础

Numpy和Pandas的综合应用——空气质量监测数据的预处理和基本分析

- 自拟新的变量

```
pd.get_dummies(data['等级'])
data.join(pd.get_dummies(data['等级']))
```

	日期	AQI	质量等级	PM2.5	PM10	SO2	CO	NO2	O3	年	季度	等级	一级优	二级良	三级轻度污染	四级中度污染	五级重度污染	六级严重污染
0	2014-01-01	81.0	良	45.0	111.0	28.0	1.5	62.0	52.0	2014	一季度	二级良	0	1	0	0	0	0
1	2014-01-02	145.0	轻度污染	111.0	168.0	69.0	3.4	93.0	14.0	2014	一季度	三级轻度污染	0	0	1	0	0	0
2	2014-01-03	74.0	良	47.0	98.0	29.0	1.3	52.0	56.0	2014	一季度	二级良	0	1	0	0	0	0
3	2014-01-04	149.0	轻度污染	114.0	147.0	40.0	2.8	75.0	14.0	2014	一季度	三级轻度污染	0	0	1	0	0	0
4	2014-01-05	119.0	轻度污染	91.0	117.0	36.0	2.3	67.0	44.0	2014	一季度	三级轻度污染	0	0	1	0	0	0

- 数据集抽取

```
np.random.seed(123)
sampler=np.random.randint(0,len(data),10)
print(sampler)
sampler=np.random.permutation(len(data))[:10]
print(sampler)

data.take(sampler)
data.loc[data['质量等级']=='优',:]
```

```
[1346 1122 1766 2154 1147 1593 1761   96   47   73]
[1883  326   43 1627 1750 1440  993 1469 1892  865]
```

	日期	AQI	质量等级	PM2.5	PM10	SO2	CO	NO2	O3	年	季度	等级
7	2014-01-08	27.0	优	15.0	25.0	13.0	0.5	21.0	53.0	2014	一季度	一级优
8	2014-01-09	46.0	优	27.0	46.0	19.0	0.8	35.0	53.0	2014	一季度	一级优
11	2014-01-12	47.0	优	27.0	47.0	27.0	0.7	39.0	59.0	2014	一季度	一级优
19	2014-01-20	35.0	优	8.0	35.0	6.0	0.3	15.0	65.0	2014	一季度	一级优
20	2014-01-21	26.0	优	18.0	25.0	27.0	0.7	34.0	50.0	2014	一季度	一级优
...	...	...	...	...	...	...	...	...	...	...	...	...
2122	2019-10-25	30.0	优	8.0	20.0	2.0	0.4	24.0	55.0	2019	四季度	一级优
2131	2019-11-03	48.0	优	33.0	48.0	3.0	0.6	34.0	33.0	2019	四季度	一级优
2135	2019-11-07	47.0	优	24.0	47.0	3.0	0.5	37.0	44.0	2019	四季度	一级优
2152	2019-11-24	30.0	优	7.0	30.0	3.0	0.2	11.0	58.0	2019	四季度	一级优
2153	2019-11-25	40.0	优	13.0	30.0	3.0	0.4	32.0	29.0	2019	四季度	一级优

387 rows × 12 columns

二、Python基础

Matplotlib库介绍

- Matplotlib库由各种可视化类构成，内部结构复杂
- matplotlib.pylot是绘制各类可视化图形的命令字库，相当于快捷方式
- 安装命令 pip install matplotlib
- 导入命令matplotlib.pylot依赖包

```
import matplotlib.pyplot as plt
```

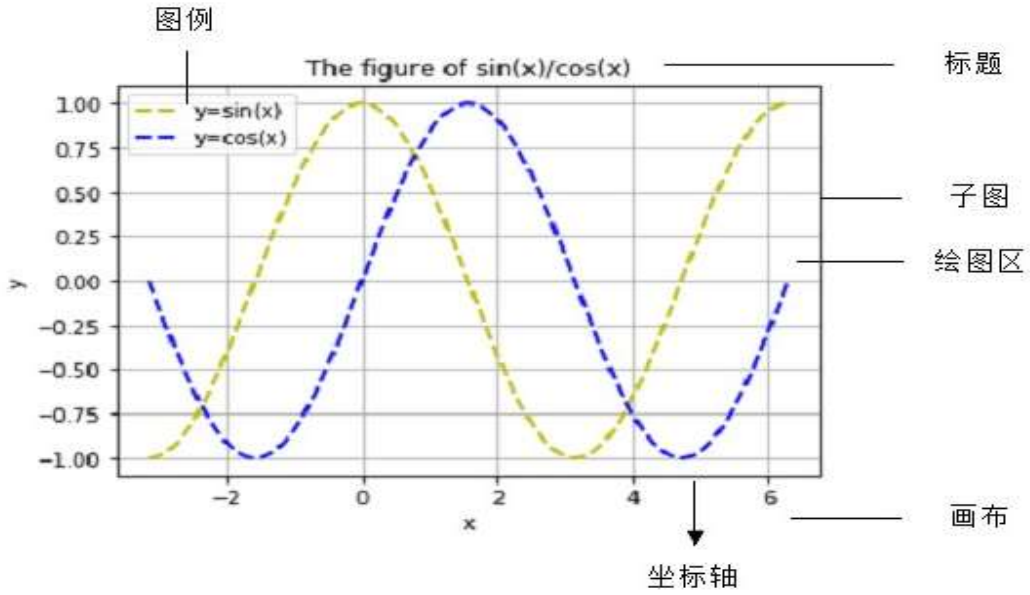
pyplot模块的常用函数表

函数	描述
figure	创建一个空白画布，可以指定画布的大小和像素
add_subplot	创建子图，可以指定子图的行数，列数和标号
subplots	建立一系列子图，返回fig,ax一个fig序列对象，建立一个axis序列
title	设置图表标题，可以指定标题的名称、颜色、字体等参数
xlabel	设置x轴名称，可以指定名称、颜色、字体等参数
ylabel	设置y轴名称，可以指定名称、颜色、字体等参数
xlim	指定x轴的刻度范围
ylim	指定y轴的刻度范围
legend	指定图例，及图例的大小、位置、标签
savefig	保存图形
show	显示图形

二、Python基础

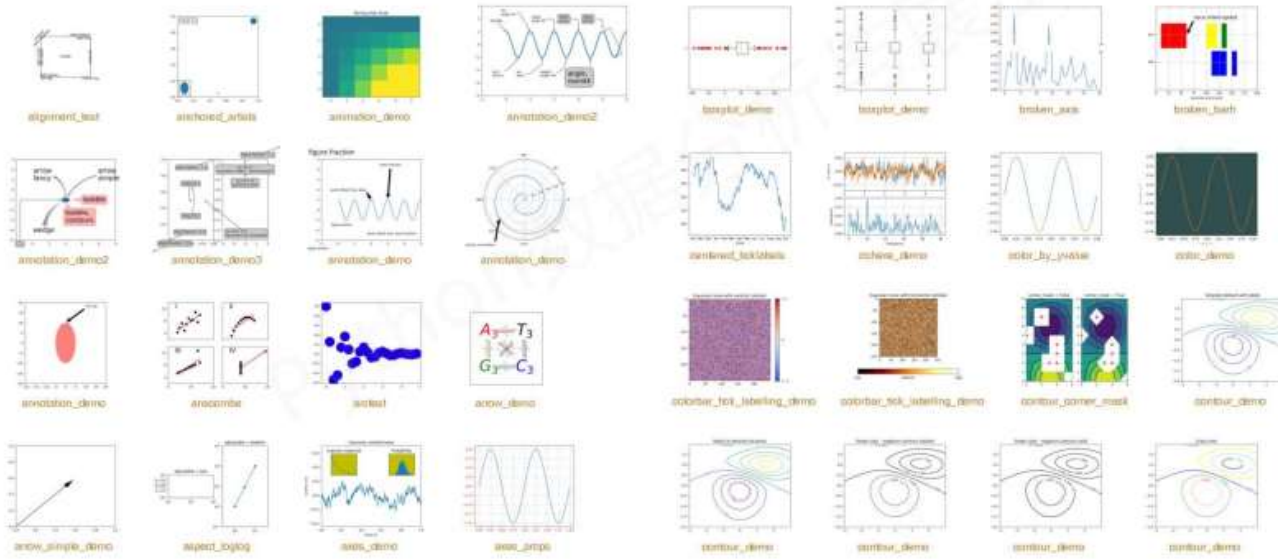
Matplotlib库介绍

图表结构一般包括：画布、图表标题、绘图区、x轴（水平轴）和y轴（垂直轴）、图例等基本元素。



Matplotlib库的效果

http://matplotlib.org/gallery.html



二、Python基础

Matplotlib的综合应用——空气质量监测数据的图形化展示

- **展示2014年至2019年每日AQI的时序变化特点**

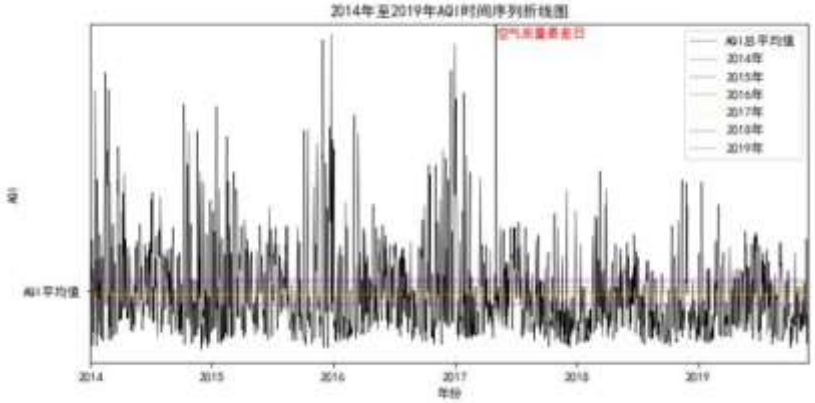
主要代码

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

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

data=pd.read_excel('北京市空气质量数据.xlsx')
data=data.replace(0,np.NaN)

plt.figure(figsize=(10,5))
plt.plot(data['AQI'],color='black',linestyle='-',linewidth=0.5)
plt.axhline(y=data['AQI'].mean(),color='red', linestyle='-',linewidth=0.5,label='AQI总平均值')
data['年']=data['日期'].apply(lambda x:x.year)
AQI_mean=data['AQI'].groupby(data['年']).mean().values
year=['2014年','2015年','2016年','2017年','2018年','2019年']
col=['red','blue','green','yellow','purple','brown']
for i in range(6):
    plt.axhline(y=AQI_mean[i],color=col[i], linestyle='--',linewidth=0.5,label=year[i])
plt.title('2014年至2019年AQI时间序列折线图')
plt.xlabel('年份')
plt.ylabel('AQI')
plt.xlim(xmax=len(data), xmin=1)
plt.ylim(ymax=data['AQI'].max(),ymin=1)
plt.yticks([data['AQI'].mean()],['AQI平均值'])
plt.xticks([1,365,365*2,365*3,365*4,365*5],['2014','2015','2016','2017','2018','2019'])
plt.legend(loc='best')
plt.text(x=list(data['AQI']).index(data['AQI'].max()),y=data['AQI'].max()-20,s='空气质量最差日',color='red')
plt.show()
```

代码说明

```
#代码说明：
(1) 第3行：Matplotlib的Pyplot子模块，指定别名为plt。
(2) 第5至7行：指定立即显示所绘图形，且通过参数设置解决图形中文显示乱码问题。
(3) 第12行：利用函数plt.figure说明图形的一般特征，如这里宽为10高5。
(4) 第13行：利用函数plt.plot绘制序列折线图（还可以绘制其他图）。同时，指定折线颜色、线形、线宽等。
(5) 第14行：利用函数plt.axhline在参数y指定的位置上画一条平行于横坐标的直线，并给定直线图例文字。
    plt.axvline可参数x指定的位置上画一条平行于纵坐标的直线。
(6) 第16至20行：首先，分组计算各年AQI的平均值；然后，通过for循环绘制多条平行于横坐标的直线，表征各年AQI平均值。
(7) 第21至23行：利用title()、xlabel()、ylabel()指定图的标题，横纵坐标的坐标标签。
(8) 第24，25行：利用xlim()、ylim()指定横纵坐标的取值范围。
(9) 第26，27行：利用xticks()、yticks()在指定坐标刻度位置上给出刻度标签。
(10) 第28行：利用legend()在指定位置（这里best表示最优位置）显示图例。
(11) 第29行：利用text()在指定的行列位置上显示指定文字
(12) 第30行：利用show()表示本次绘图结束。
```

运行结果



二、Python基础

Matplotlib的综合应用——空气质量监测数据的图形化展示

- **AQI的分布特征及相关性分析**

1）绘出线图展示2014年至2019年的年均AQI的变化特点；

2）绘出直方图展示2014年至2019年AQI的整体分布特征；

3）绘出散点图展示AQI和PM2.5的相关性；

4）绘出饼图展示空气质量等级的分布特征。

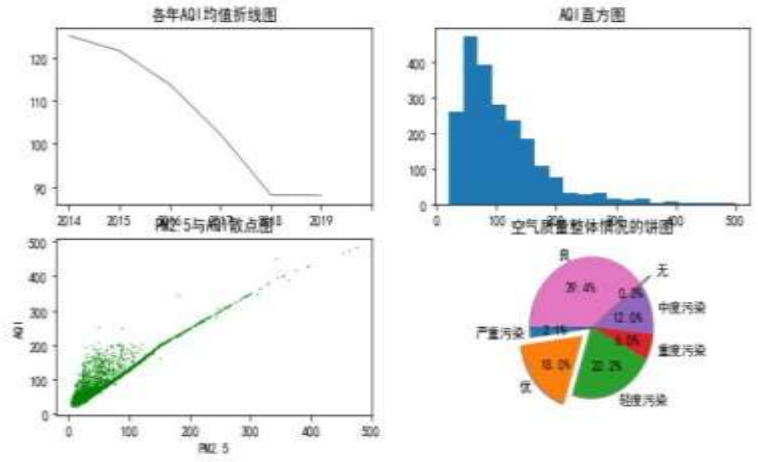
主要代码

```
import warnings
warnings.filterwarnings(action = 'ignore')
plt.figure(figsize=(10,5))
plt.subplot(2,2,1)
plt.plot(AQI_mean,color='black',linestyle='-',linewidth=0.5)
plt.title('各年AQI均值折线图')
plt.xticks([0,1,2,3,4,5,6],['2014','2015','2016','2017','2018','2019'])
plt.subplot(2,2,2)
plt.hist(data['AQI'],bins=20)
plt.title('AQI直方图')
plt.subplot(2,2,3)
plt.scatter(data['PM2.5'],data['AQI'],s=0.5,c='green',marker='.')
plt.title('PM2.5与AQI散点图')
plt.xlabel('PM2.5')
plt.ylabel('AQI')
plt.subplot(2,2,4)
tmp=pd.value_counts(data['质量等级'],sort=False)  #等同：tmp=data['质量等级'].value_counts()
share=tmp/sum(tmp)
labels=tmp.index
explode = [0, 0.2, 0, 0, 0,0.2,0]
plt.pie(share, explode = explode,labels = labels, autopct = '%3.1f%%',startangle = 180, shadow = True)
plt.title('空气质量整体情况的饼图')
```

代码说明

```
#代码说明：
(1) 第1，2行：导入warnings模块，并指定忽略代码运行过程中的警告信息。
(2) 第4行：subplot(2,2,1)表示将绘图区域分成2行2列4个单元，且下一副图将在第1个单元显示。
(3) 第8行：subplot(2,2,2)表示将绘图区域分成2行2列4个单元，且下一副图将在第2个单元显示。
(4) 第9行：利用hist()绘制AQI的直方图，图中包含20个柱形条，即将数据分成20组。
(5) 第12行：利用scatter()绘制PM2.5和AQI的散点图。并指定点的大小（s），颜色（c）和形状（marker）。
(6) 第21行：利用pie()绘制饼图。
```

运行结果



二、Python基础

Matplotlib的综合应用——空气质量监测数据的图形化展示

- **AQI的分布特征及相关性分析**

5）由于上面四幅图出现了重叠现象，因此采用以下方式对图形进行优化调整。

主要代码

```
fig,axes=plt.subplots(nrows=2,ncols=2,figsize=(10,5))
axes[0,0].plot(AQI_mean,color='black',linestyle='-',linewidth=0.5)
axes[0,0].set_title('各年AQI均值折线图')
axes[0,0].set_xticks([0,1,2,3,4,5,6])
axes[0,0].set_xticklabels(['2014','2015','2016','2017','2018','2019'])
axes[0,1].hist(data['AQI'],bins=20)
axes[0,1].set_title('AQI直方图')
axes[1,0].scatter(data['PM2.5'],data['AQI'],s=0.5,c='green',marker='.')
axes[1,0].set_title('PM2.5与AQI散点图')
axes[1,0].set_xlabel('PM2.5')
axes[1,0].set_ylabel('AQI')
axes[1,1].pie(share, explode = explode,labels = labels, autopct = '%3.1f%%',startangle = 180, shadow = True)
axes[1,1].set_title('空气质量整体情况的饼图')
fig.subplots_adjust(hspace=0.5)
fig.subplots_adjust(wspace=0.5)
```

代码说明

#代码说明：

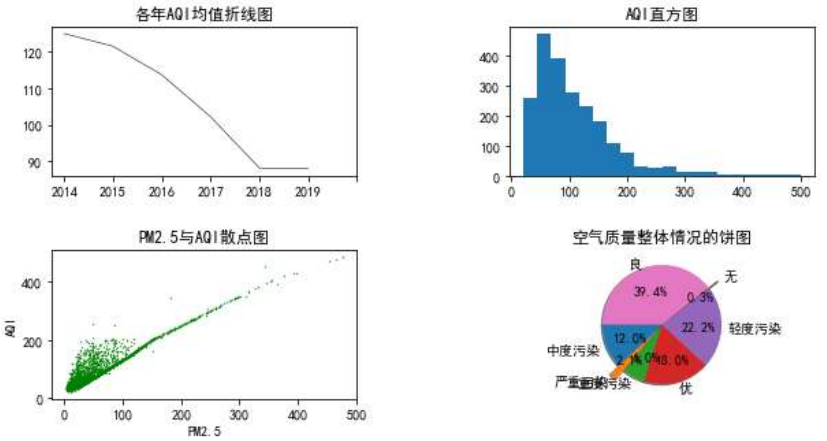
（1）第1行：说明绘图区域的宽和高，并指定将绘图区域分成2行2列4个单元。结果将赋值给fig和axes对象。可通过fig对整个图的特征进行设置，axes对应各个单元格对象。

（2）通过图形单元索引的方式指定绘图单元。例如：axes[0,0]表示第1行第1列的单元格。

（3）单元格对象的图标题、坐标轴标签、坐标刻度等，需采用set_title()、set_xlabel()、set_ylabel()、set_xticks()、set_xticklabels()设置。

（4）第14，15行：利用subplots_adjust调整各图形单元行或列之间的距离。

优化后的结果



二、Python基础

Sklearn库介绍

Sklearn (scikit-learn) 是基于 Python 语言的机器学习工具。在 sklearn 里面有六大任务模块分别是：

- Classification 分类　　Regression 回归
- Clustering 聚类　　Dimensionality reduction 数据降维
- Model Selection 模型选择　　Preprocessing 数据与处理

- 功能：只需要调用几行sklearn 库里的API即可实现一个复杂的机器学习算法
- 安装命令 pip install scikit-learn
- **更多学习请参考**scikit-learn**官网：** https://scikit-learn.org/stable/

二、Python基础

Sklearn示例——使用感知机模型实现鸢尾花分类

导入必需库

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.linear_model import Perceptron
from sklearn.model_selection import train_test_split
```

主要代码

```
#加载数据
iris = load_iris()
# 将鸢尾花4个特征，以4列存入pandas的数据框架
df = pd.DataFrame(iris.data, columns=iris.feature_names)

# 在最后一列追加 加入（目标值）列数据
df['label'] = iris.target

# 选取数据，前100行，前两个特征，最后一列的目标值
data = np.array(df.iloc[:100, [0, 1, -1]])
# 生成感知机的标签值，+1， -1，第一种 - 1，第二种 + 1
for i in range(len(data)):
    if data[i,-1] == 0:
        data[i,-1] = -1

#数据分割
# X是除最后一列外的所有列，y是最后一列
X, y = data[:, :-1], data[:, -1]

# 调用sklearn的train_test_split方法，将数据随机分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X,   #被划分的样本特征集
                                                    y,   #被划分的样本目标集
                                                    test_size=0.3, #测试样本占比
                                                    random_state=1) #随机数种子

# 定义感知机
clf = Perceptron(fit_intercept=False, max_iter=1000, shuffle=False)

# 使用训练数据进行训练
clf.fit(X_train, y_train)

#评价模型
print(clf.score(X_test, y_test))
```

运行结果

```
0.9666666666666667
```

三、数据说明

- 数据文件（**ReportCard1.txt**和**ReportCard2.txt**）分别记录着60名学生的学号、性别以及8门课程的成绩，如下图所示：

ReportCard1 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

id	sex	politics	chinese	math
92103	2.00	NA	NA	NA
92239	2.00	40.00	63.00	44.00
92142	2.00	NaN	70.00	59.00
92223	1.00	56.00	91.00	65.50
92144	1.00	59.00	79.00	34.00
92217	2.00	60.00	82.50	76.50
92111	1.00	61.00	86.00	74.00
92146	1.00	61.00	69.00	45.00
92234	1.00	66.00	79.00	55.50
92113	1.00	70.00	85.00	66.00
92126	1.00	70.00	92.00	56.00
92211	2.00	71.00	73.00	69.00
92226	1.00	73.00	77.00	52.50

ReportCard2 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

id	biology	physics	chemistry	geography	history
92146	20.00	49.00	32.00	32.00	51.00
92239	21.00	54.00	26.00	26.00	55.00
92142	22.00	68.00	26.00	26.00	63.00
92141	28.00	69.00	43.00	43.00	74.00
92236	28.00	63.00	36.00	36.00	52.00
92232	30.00	60.00	43.00	61.50	79.00
92134	31.00	67.00	41.00	72.50	77.00
92145	33.00	64.00	34.00	34.00	71.00
92144	34.00	57.00	37.00	37.00	76.00
92217	35.00	81.00	60.00	70.50	74.00
92127	38.00	76.00	47.00	79.00	84.00
92209	39.00	89.00	53.00	68.50	81.00
92226	39.00	81.00	44.00	78.00	73.00

注意：1、两个文件里学号id是乱序排列，但学生学号是唯一且对应的，都是60个；
2、sex里面1表示男性，2表示女性
3、NA表示为缺失值，自行处理。

四、实验任务

➢ 1、Python编程：学生成绩数据的处理和基本分析。

请将学生成绩数据读入Pandas数据框，并做如下预处理和基本分析：

1）将两个数据文件按学号合并为一个数据文件，得到包含所有课程成绩的数据文件；

2）计算每个同学所有课程的总成绩和平均成绩；

3）将数据按总成绩的降序排序；

4）按性别分别计算各门课程的平均成绩；

5）按优、良、中、及格和不及格，对2）内的平均成绩进行分组；

6）按性别统计优、良、中、及格和不及格的人数。

等级	分值
优	[90, 100]
良	[80, 90)
中	[70, 80)
及格	[60, 70)
不及格	[0, 60)

➢ 2、Python编程：学生成绩数据的图形化展示。

对以上两份数据文件，做如下图形化展示：

1）绘制总成绩的直方图；

2）绘制平均成绩的优、良、中、及格和不及格的饼图；

3）绘制总成绩和数学（math）成绩的散点图。

注意：重要代码处要添加注释！！！

五、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-综合班**」课程 - 实验一；
- 3、提交截止时间：下周二 5月21号 晚24点；
- 4、文件夹&压缩包命名要求： 学号_姓名_统计机器学习实验一
- 5、提交内容：实验截图文档(pdf格式)，包含两个任务运行结果截图和实验代码（文本式）。

其他：

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。

统计机器学习实验

同学们，请开始实验吧！