实验二　Numpy 数值计算

1.实验类型

验证型实验

2.实验目的和要求

（1）掌握 Numpy 数组对象的创建；

（2）掌握 Numpy 数组对象的索引与变换；

（3）掌握 Numpy 中读/写文件的方法；

（4）能利用 Numpy 进行简单的统计分析；

3.实验内容

题目一：创建数组并计算

(1) 创建一个数值范围为 0-1，间隔为 0.01 的数组(100 个元素)。

(2) 创建 100 个服从正态分布的随机数。

(3) 读取 stu.txt,文件中每行为某同学各科目的成绩，请利用 numpy 中的统计函数进行以下统计：求出每位同学的各科目平均成绩；求出每个科目的最高分与最低分。

题目二：电子商务数据统计分析

3.1 项目介绍：

随着社会经济的发展，人们对服装的需求越来越多样化，而女性服装的变化也成为值得分析的一个关注点。这个项目的目的是对女性的服装进行分析，以了解客户对女性服装的态度。

注：本题目需参考给定模板来完成，在要求部分，填入代码。使用向量化的操作来完成。

3.2 数据集分析

项目的数据集内容主要是客户对服装的评论，数据集包括 23486 行和 10 个特征变量（列数，不包含索引列），而每一行包含了一个客户对服装的评论和其他相关信息。

- Clothing ID：服装的唯一 ID 号
- Age：评论者的年龄
- Title：评论标题
- Review：　评论内容
- Rating：评论员对服装的评级，从 1 到 5，1 最差，5 最好

- Recommended IND：服装是否被评论家推荐，推荐为 1，不推荐为 0
- Positive Feedback Count：正反馈计数
- Division Name：服装高级分类的分类名称
- Department Name：服装部门名称的分类名称
- Class Name：服装分类名称

3.3 分析流程：

(1)获得要分析的数据集：原数据集的每一行都是一个客户的评论，现在根据给定列 Clothing ID,Recommended IND,Positive Feedback Count,Class Name 获得需要的数据集

(2)获得唯一 Clothing ID 的列表：由于 Clothing ID 的评论数不同，我们希望统计 Clothing ID 的评论数，仅对评论数大于 400 的唯一 Clothing ID 的数据进行分析

(3)该服装的受欢迎程度：基于 Clothing ID 的列表，计算每个 Clothing ID 的评论次数和推荐次数，进而计算推荐次数占评论次数的比例，该占比即服装的受欢迎程度。

(4)计算每个 Clothing ID 的正反馈次数：对每个 Clothing ID 代表的服装进行统计，计算正反馈次数的加和，了解客户对服装的正面评价情况

(5)执行 main 方法，打印每个 Clothing ID 的统计数据

4.实验背景知识

NumPy(Numerical Python)是高性能科学计算和数据分析的基础包。它是本课介绍的几乎所有高级工具的构建基础。

（1）数组创建

*numpy.**array**(object, dtype=None, copy=True, order='K',subok=False, ndmin=0)*

参数名称	说明
object	接收array。表示想要创建的数组。无默认。
dtype	接收data-type。表示数组所需的数据类型。如果未给定，则选择保存对象所需的最小类型。默认为None。
ndmin	接收int。指定生成数组应该具有的最小维数。默认为None。

（2）索引与切片

- 一维数组的索引与 Python 的列表索引功能相似
- 多维数组的索引

- arr[r1:r2, c1:c2]
- arr[1,1] 等价 arr[1][1]
- [:] 代表某个维度的数据

（3）数组常用属性

属性	说明
ndim	返回 int。表示数组的维数
shape	返回 tuple。表示数组的尺寸，对于 n 行 m 列的矩阵，形状为(n,m)
size	返回 int。表示数组的元素总数，等于数组形状的乘积
dtype	返回 data-type。描述数组中元素的类型
itemsize	返回 int。表示数组的每个元素的大小（以字节为单位）。

（4）数组类型转换 astype()

```
new_a = a.astype(new_type)
```

```
In [119]: a = np.ones((2,3,4), dtype=np.int)

In [120]: a
Out[120]:
array([[[1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1]],

       [[1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1]]])
```

```
In [121]: b = a.astype(np.float)

In [122]: b
Out[122]:
array([[[ 1.,  1.,  1.,  1.],
        [ 1.,  1.,  1.,  1.],
        [ 1.,  1.,  1.,  1.]],

       [[ 1.,  1.,  1.,  1.],
        [ 1.,  1.,  1.,  1.],
        [ 1.,  1.,  1.,  1.]]])
```

astype()方法一定会创建新的数组（原始数据的一个拷贝），即使两个类型一致

（5）ndarray 数组向列表转换 tolist()

```
ls = a.tolist()
```

```
In [128]: a = np.full((2,3,4), 25, dtype=np.int32)

In [129]: a
Out[129]:
array([[[25, 25, 25, 25],
        [25, 25, 25, 25],
        [25, 25, 25, 25]],

       [[25, 25, 25, 25],
        [25, 25, 25, 25],
        [25, 25, 25, 25]]])

In [130]: a.tolist()
Out[130]:
[[[25, 25, 25, 25], [25, 25, 25, 25], [25, 25, 25, 25]],
 [[25, 25, 25, 25], [25, 25, 25, 25], [25, 25, 25, 25]]]
```

（6）Numpy 读/写文件

- savetxt 函数是将数组写到某种分隔符隔开的文本文件中。
  np.savetxt("../tmp/arr.txt", arr, fmt="%d", delimiter=",")
- loadtxt 函数执行的是把文件加载到一个二维数组中。
  np.loadtxt("../tmp/arr.txt",delimiter=",")
- genfromtxt 函数面向的是结构化数组和缺失数据。
  np.genfromtxt("../tmp/arr.txt", delimiter = ",")

(7) Python 日期和时间模块 datetime

python 中时间日期格式化符号：

- %y 两位数的年份表示（00-99）
- %Y 四位数的年份表示（000-9999）
- %m 月份（01-12）
- %d 月内中的一天（0-31）
- %H 24 小时制小时数（0-23）
- %I 12 小时制小时数（01-12）
- %M 分钟数（00=59）
- %S 秒（00-59）
- %a 本地简化星期名称
- %A 本地完整星期名称
- %b 本地简化的月份名称
- %B 本地完整的月份名称
- %c 本地相应的日期表示和时间表示
- %j 年内的一天（001-366）
- %p 本地 A.M.或 P.M.的等价符
- %U 一年中的星期数（00-53）星期天为星期的开始
- %w 星期（0-6），星期天为星期的开始
- %W 一年中的星期数（00-53）星期一为星期的开始
- %x 本地相应的日期表示
- %X 本地相应的时间表示
- %Z 当前时区的名称
- %% %号本身

（8）去重数据：unique 方法

通过 unique 函数可以找出数组中的唯一值并返回已排序的结果。

```
numpy.unique(arr, return_index, return_inverse, return_counts)
```

参数说明：

arr：输入数组，如果不是一维数组则会展开

return_index：如果为 true，返回新列表元素在旧列表中的位置（下标），并以列表形式储

return_inverse：如果为 true，返回旧列表元素在新列表中的位置（下标），并以列表形式储

return_counts：如果为 true，返回去重数组中的元素在原数组中的出现次数

（9）Numpy 常用统计函数

- 求均值：np.mean()，求和：np.sum()，
- 求最大：np.max()，求最小：np.min()
- 求标准差：np.std()，求方差：np.var()
- 求最大值的索引：np.argmax()，求最小值的索引：np.argmin()
- 求累加：np.cumsum()，求累乘：np.cumprod()
- 是否所有元素满足条件：np.all()
- 是否至少一个元素满足条件：np.any()

5.实验思考

（1）NumPy 从这个库的名字理解，这个库的作用是什么？

（2）怎样取出数组内部的某个元素？