|  |
| --- |
| **计算机专业类课程** |
| **实验报告** |
| **课程名称：数据挖掘与大数据分析**  **学　　院：计算机科学与工程学院**  **专　　业：互联网加**  **学生姓名：李宏洋**  **学　　号：2020080301012**  **指导教师：曾伟** |
|  |
| **日　　期：　　　2022 年　3　月　5　日** |

目录

[**一、 实验名称：认识数据与数据预处理** 2](#_Toc97470464)

[**二、 实验学时：4** 2](#_Toc97470465)

[**三、 实验内容和目的：** 2](#_Toc97470466)

[（一） 实验内容 2](#_Toc97470467)

[（二） 实验目的 2](#_Toc97470468)

[**四、 实验数据及结果分析：** 2](#_Toc97470469)

[（一） 归一化 2](#_Toc97470470)

[1、 思路介绍 2](#_Toc97470471)

[2、 实验结果展示与分析 3](#_Toc97470472)

[3、 完整代码展示 4](#_Toc97470473)

[（二） 缺失值处理 5](#_Toc97470474)

[1、 思路介绍 5](#_Toc97470475)

[2、 实验结果展示与分析 6](#_Toc97470476)

[3、 完整代码展示 7](#_Toc97470477)

[（三） 特征筛选 8](#_Toc97470478)

[1、思路介绍 8](#_Toc97470479)

[2、实验结果展示与分析 8](#_Toc97470480)

[3、完整代码展示 9](#_Toc97470481)

[（四） 可视化 10](#_Toc97470482)

[1、思路介绍 10](#_Toc97470483)

[2、实验结果展示与分析 10](#_Toc97470484)

[3、完整代码展示 12](#_Toc97470485)

[**五、 实验结论、心得体会和改进建议：** 12](#_Toc97470486)

**电 子 科 技 大 学**

**实 验 报 告**

**实验一**

1. **实验名称：认识数据与数据预处理**
2. **实验学时：4**
3. **实验内容和目的：**
4. 实验内容
5. 安装python与anaconda
6. 对数据进行归一化
7. 对数据进行缺失值处理
8. 对数据进行特征筛选
9. 对数据进行可视化处理
10. 实验目的
11. 掌握对数据进行诸如归一化、缺失值处理、特征筛选以及可视化等预处理的方法
12. 进一步熟悉python的使用，并熟练使用python中关于数据处理的第三方库
13. **实验数据及结果分析：**
    1. 归一化
14. 思路介绍

首先我们分析归一化的要求：我们需要对存在iris.txt文件中的csv数据进行处理并将结果存至iris\_normal.txt文件中。为了达成以上目标我们需要进行三步操作：首先，读取iris.txt文件并将其中的数据格式改变为方便处理的格式；其次，利用公式进行归一化处理；最后，将处理结果储存进iris\_normal.txt中。

1. 数据读取

由于数据为csv结构的数据，所以这里使用pandas库中的read\_csv函数进行读取。同时，由于数据最后一列为我们不需要处理的字符串，故我们用drop方法去掉数据矩阵的最后一列以方便后续处理。

1. 归一化原理

由于这些数据均是连续的数值数据，所以采取最小-最大规范化方法进行归一化：

经过归一化之后的数据均统一至[0,1]区间。这里再具体处理的时候未来防止分母为0增加了一个拉普拉斯因子。

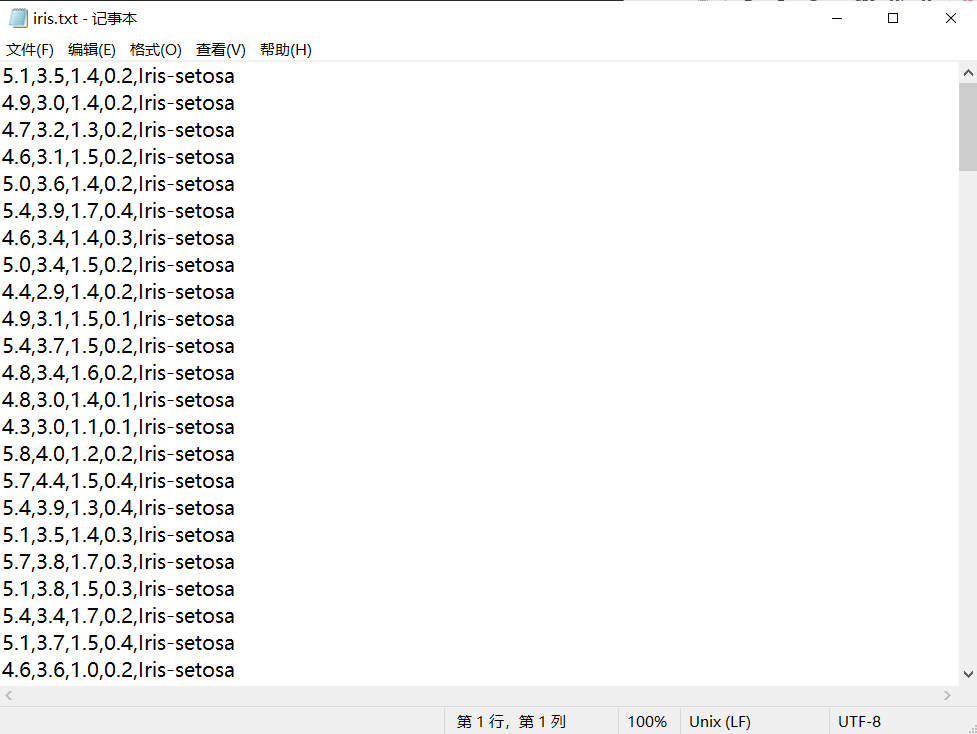
1. 储存结果

处理好数据后，我们通过pandas提供的DataFrame方法对数据进行重新组合，并将开始提取出来的字符串用insert方法放回。最后用to\_csv方法将归一化后的数据写入指定路径的txt文件中。

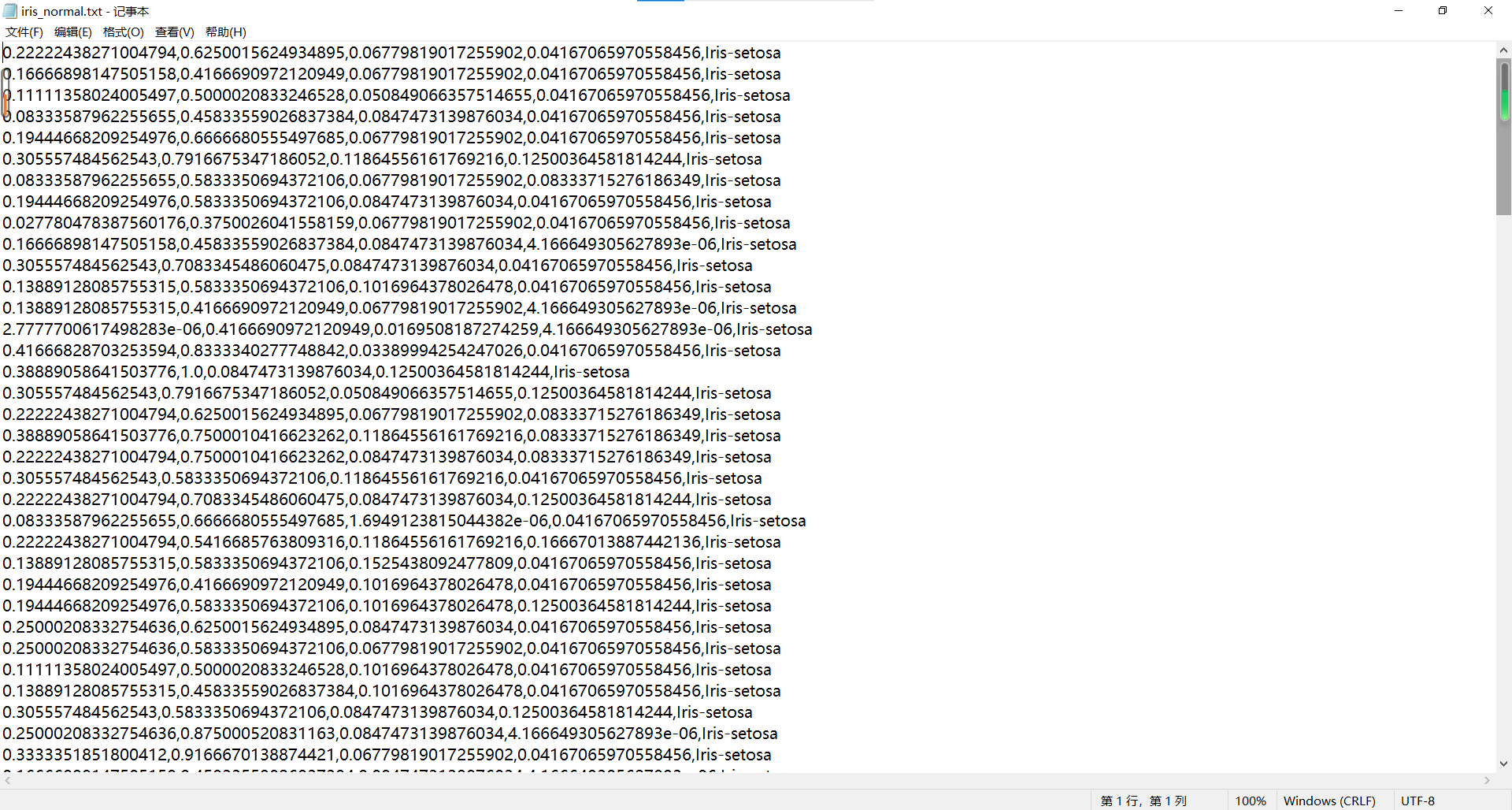
1. 实验结果展示与分析

（1）实验结果展示

实验前的数据储存于iris.txt中，如下图所示：：



实验后的数据存储于iris\_normal.txt中，如下图所示：



（2）实验结果分析

单从结果来看的话，完全符合预期，将数据均归一化至[0,1]区间内。从结果来看，在设计程序时将存入到数据进行位数上的保留会让结果美观一些，但是最后从精度考虑没有采取这种方法进行保留。

1. 完整代码展示

**import** numpy **as** np  
**import** pandas **as** pd  
  
*# 读取数据(Iris)***def** loadIris(address):  
 spf = pd.read\_csv(address, sep=**','**, index\_col=**False**, header=**None**)  
 strs = spf[4]  
 *# 用drop方法去掉最后一列字符串方便处理* spf.drop([4], axis=1, inplace=**True**)  
 **return** spf.values, strs  
  
  
*# 归一化***def** normalization(data\_matrix):  
 e = 1e-5 *# 防止出现0，加一个拉普拉斯因子* **for** c **in** range(4):  
 maxNum = np.max( data\_matrix[:, c])  
 minNum = np.min( data\_matrix[:, c])  
 *# 利用最小-最大规范化公式进行归一化，统一至[0，1]区间* data\_matrix[:, c] = (data\_matrix[:, c] - minNum + e)/(maxNum - minNum + e)  
 **return** data\_matrix  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 fielpath = **'./data/iris.txt'** writepath = **'./data/ans/iris\_normal.txt'** *# read data* data\_matrix, str\_name = loadIris(fielpath)  
 *# print(data\_matrix)  
 # normalization* data\_matrix = normalization(data\_matrix)  
  
 *# 重新组合数据（标签无法归一化）* spf = pd.DataFrame(data\_matrix)  
  
 *# 保存归一化后的数据* strs = str\_name.values  
 spf.insert(4, 4, strs)  
 spf.to\_csv(writepath, index=**False**, header=**False**)

* 1. 缺失值处理

1. 思路介绍

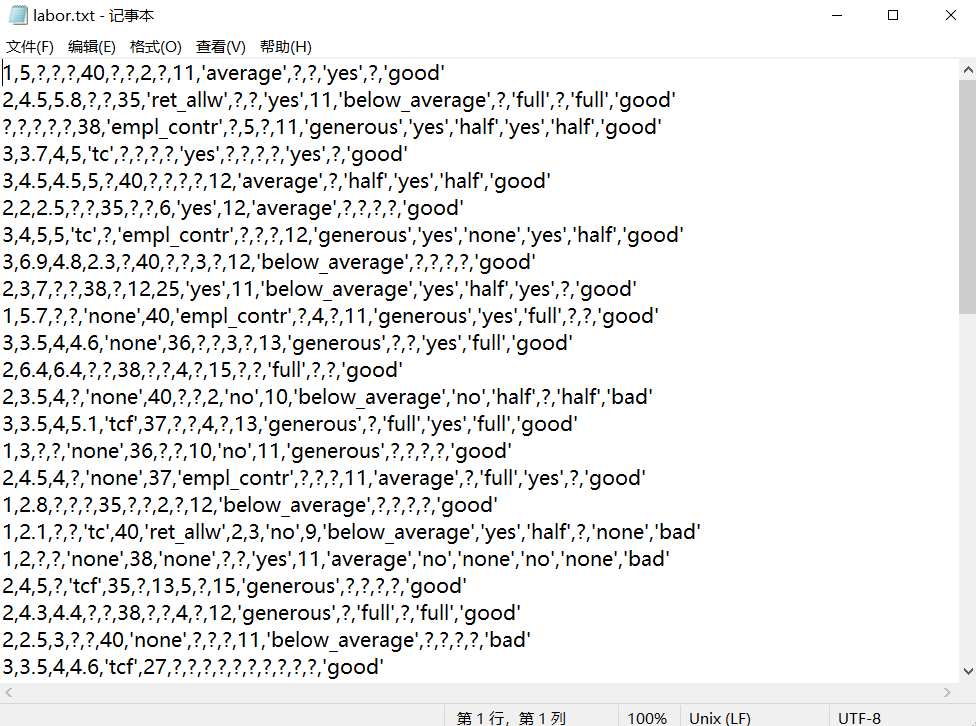
分析缺失值处理项目的具体要求，可以看到，这部分实验由类似前面的三个部分组成：读取、缺失值处理、存入。读取与存入的处理方式与之前的类似，这里重点介绍缺失值处理的逻辑。

观察数据，不难看见缺失数据均由字符“？”占位，所以我们围绕这个展开处理。

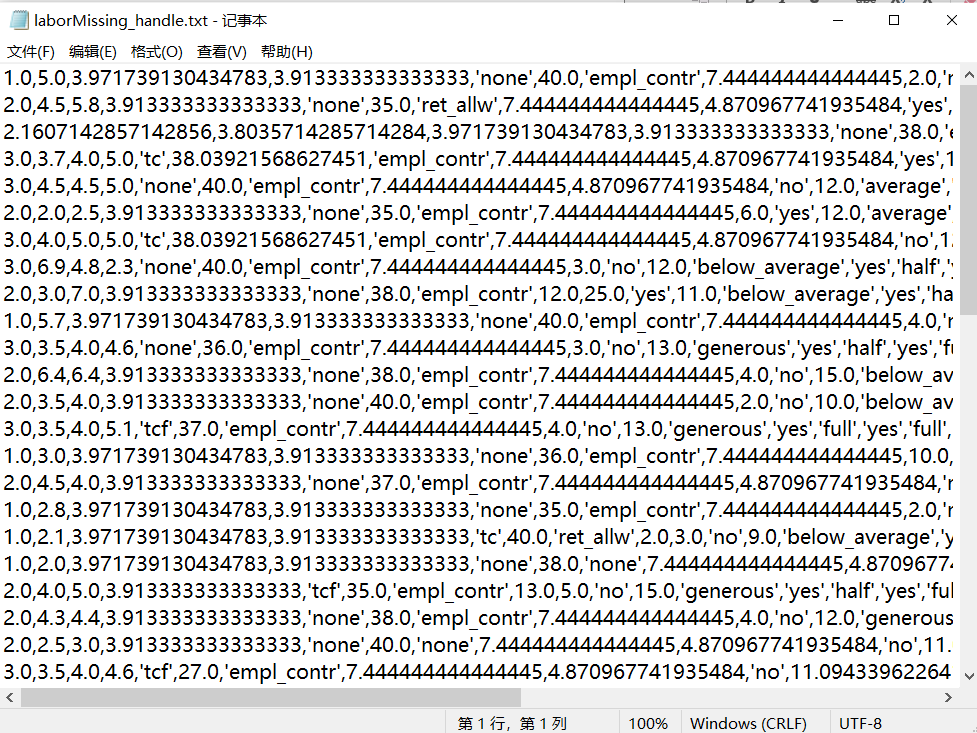
首先，利用pandas提供的colums方法给数据加上表头，以便于后续对不同类型的数据采取对应方式处理。总的来说，可以将数据分为两类，一种是数值类型的，另一种是标称类型的，也即字符串。对于数值型数据，我们用该列所有非‘？’位置上的平均值来补充。对于标称类型的数据，我们用在该列出现最频繁的数据填充。

1. 实验结果展示与分析
2. 实验结果展示

实验前的数据存储于labor.txt中，如下图所示：



实验后的数据存储于laborMissing\_handle.txt中，如下图所示：



1. 实验结果分析

从结果来看完全符合预期，对于矩阵的处理没有任何出入。标称数据与数值数据均得到了正确的处理。

1. 完整代码展示

**import** numpy **as** np  
**import** pandas **as** pd  
**from** collections **import** Counter  
  
  
**def** loadLabor(address):  
 spf = pd.read\_csv(address, sep=**','**, index\_col=**False**, header=**None**)  
 column = [**'duration'**, **'wage-increase-first-year'**, **'wage-increase-second-year'**, **'wage-increase-third-year'**,  
 **'cost-of-living-adjustment'**, **'working-hours'**, **'pension'**,**'standby-pay'**,  
 **'shift-diffrential'**, **'education-allowance'**, **'statutory-holidays'**,**'vacation'**,  
 **'longterm-disability-assistance'**, **'contribution-to-dental-plan'**, **'bereavement-assistance'**,  
 **'contribution-to-health-plan'**, **'class'**]  
 spf.columns = column  
  
 *# label data* str\_typename = [**'cost-of-living-adjustment'**, **'pension'**, **'education-allowance'**,  
 **'vacation'**, **'longterm-disability-assistance'**, **'contribution-to-dental-plan'**,  
 **'bereavement-assistance'**, **'contribution-to-health-plan'**, **'class'**]  
 str2numeric = {}  
 str2numeric[**'?'**] = **'-1'** spf = spf.replace(str2numeric)  
 **return** spf, str2numeric, str\_typename  
  
  
**def** fillMissData(spf, str2numeric):  
 row, col = spf.shape  
 columns = spf.columns  
 **for** column\_name **in** columns:  
 **if** column\_name **not in** str2numeric:  
 *# number,first strategy* tmp = spf[column\_name].apply(float)  
 ave = np.average(tmp[tmp != -1])  
 tmp[tmp == -1] = ave  
 spf[column\_name] = tmp  
 **else**:  
 *# label,second strategy* v = spf[column\_name].values  
 v1 = v[v != **'-1'**]  
 c = Counter(v1)  
 cc = c.most\_common(1)  
 v[v == **'-1'**] = cc[0][0]  
 **return** spf  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 fielpath = **'./data/labor.txt'** fillFilepath = **'./data/ans/laborMissing\_handle.txt'** spf, str2numeric1, str2numeric = loadLabor(fielpath)  
 spf = fillMissData(spf, str2numeric)  
 spf.to\_csv(fillFilepath, index=**False**, header=**False**)

* 1. 特征筛选

1、思路介绍

分析这部分项目要求，我们需要对数据矩阵的特征进行筛选，并按照信息增益递增的顺序输出各信息增益与特征。

文件存取不做赘述。关于信息增益的计算我们只需要按照公式求解即可：

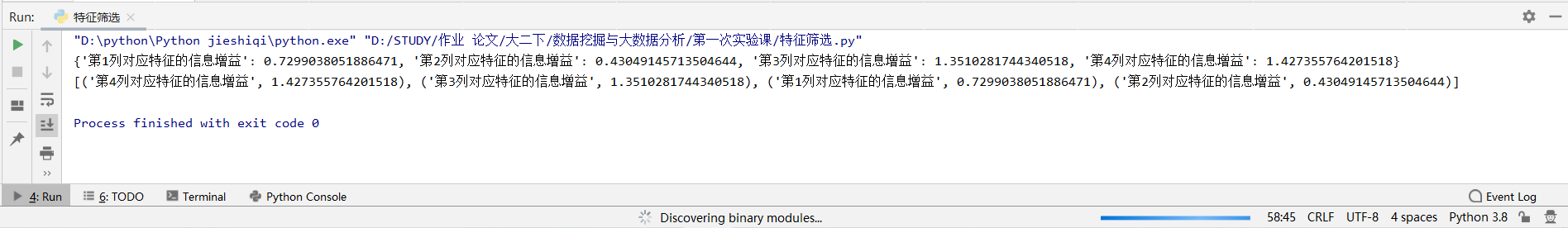
值得注意的是，我在老师给出的代码基础上为了实现特征的输出引入了一个字典类型变量参与处理。

2、实验结果展示与分析

（1）实验结果展示

数据与归一化部分一致，存于iris.txt中。

运行结果如下：



第一行为处理后的字典类型变量，第二行为根据值的大小进行排序之后的输出。

1. 实验结果分析

各行的信息增益均被合理计算在正确范围内。通过运算数据我们可知各列对应特征的信息增益由大到小为：第四列、第三列、第一列、第二列。

3、完整代码展示

**import** numpy **as** np  
**import** pandas **as** pd  
**from** collections **import** Counter  
  
  
**def** loadIris(address):  
 spf = pd.read\_csv(address, sep=**','**, index\_col=**False**, header=**None**)  
 strs = spf[4]  
 spf.drop([4], axis=1, inplace=**True**)  
 **return** spf.values, strs  
  
  
**def** featureSelection(features, label):  
 featureLen = len(features[0,:])  
 label\_count = Counter(label)  
 samples\_energy = 0.0  
 data\_len = len(label)  
 **for** i **in** label\_count.keys():  
 label\_count[i] /= float(data\_len)  
 samples\_energy -= label\_count[i] \* np.log2(label\_count[i])  
 informationGain = []  
 info = {}  
 **for** f **in** range(featureLen):  
 af = features[:, f]  
 minf = np.min(af)  
 maxf = np.max(af) + 1e-4  
 width =(maxf - minf)/10.0  
 d = (af - minf)/width  
 dd = np.floor(d)  
 c = Counter(dd)  
 sub\_energy = getEnergy(c, dd, label)  
 informationGain.append(samples\_energy - sub\_energy)  
 info[**'第{}列对应特征的信息增益'**.format(f+1)] = samples\_energy - sub\_energy  
  
 **return** informationGain, info  
  
  
**def** getEnergy(c, data, label):  
 dataLen = len(label)  
 energy = 0.0  
 **for** key, value **in** c.items():  
 c[key] /= float(dataLen)  
 label\_picked = label[data == key]  
 l = Counter(label\_picked)  
 e = 0.0  
 **for** k, v **in** l.items():  
 r = v/float(value)  
 e -= r \* np.log2(r)  
 energy += c[key] \* e  
 **return** energy  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 fielpath = **'./data/iris.txt'** data\_matrix, str\_name = loadIris(fielpath)  
 informationGain,info = featureSelection(data\_matrix, str\_name.values)  
 print(info)  
 d\_order = sorted(info.items(), key=**lambda** x: x[1], reverse=**True**)  
 print(d\_order)  
 *# print(informationGain)*

* 1. 可视化

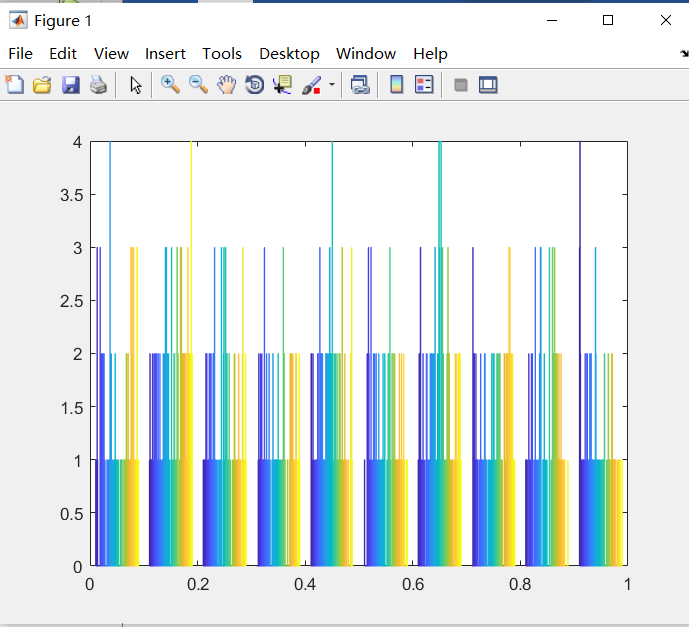
1、思路介绍

本部分实验主要是通过matlab实现数据的可视化，掌握可视化过程。由于matlab中均将此部分功能用函数封装好了，故而主要看代码部分。

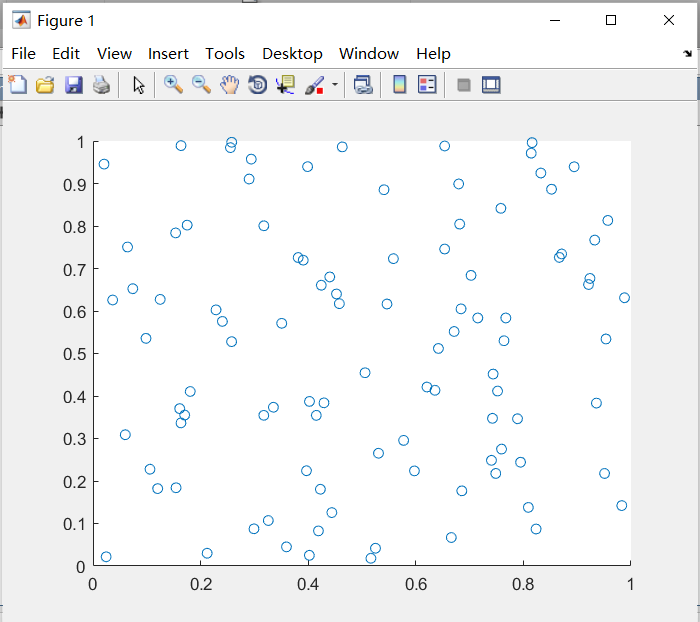
2、实验结果展示与分析

（1）实验结果展示

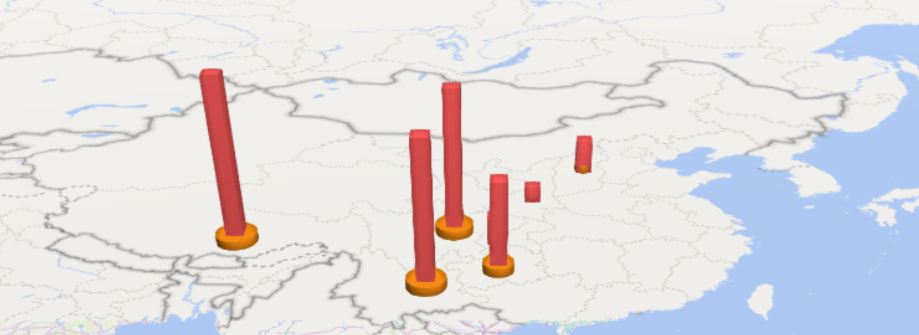
直方图可视化如下：



散点图可视化如下：



同时我发现excel也可以通过三维地图功能实现可视化，图形精美，特此展示，具体如下，第一张图为数据，第二张图为可视化图片。



3、完整代码展示

直方图代码：

function myfun

a = rand(9,100);

hist(a);

end

散点图代码：

function myfun1

a = rand(1,100);

b = rand(1,100);

scatter(a,b);

end

三维地图为图形界面操作，没有代码。

1. **实验结论、心得体会和改进建议：**

对于本次实验，由于我有一定python基础，没有像其他同学一样无措，对老师给的标准源码也有一定基于自己理解的修改。同时，老师讲解仔细，助教也耐心认真，体验很好。通过本次实验，我不仅进一步巩固并实现了课堂上讲授的算法，还对知识有了自己的理解。