**广州大学学生实验报告**

**开课学院及实验室：**计算机科学与网络工程学院软件实验室 **2020年 1 月1日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | **计算机科学与网络工程学院** | **年级/专业/班** | **软件183** | **姓名** | 张世华 | **学号** | 1806300024 |
| **实验课程名称** | **机器学习与数据挖掘实验** | | | | | **成绩** |  |
| **实验项目名称** | 数据统计与可视化 | | | | | **指导老师** | 彭伟龙 |

**实验二**

**一、实验题目**

基于实验一中清洗后的数据练习统计和视化操作，100个同学（样本），每个同学有11门课程的成绩（11维的向量）；那么构成了一个100x11的数据矩阵。以你擅长的语言C/C++/Java/Python/Matlab，编程计算：

1. 请以课程1成绩为x轴，体能成绩为y轴，画出散点图。
2. 以5分为间隔，画出课程1的成绩直方图。
3. 对每门成绩进行z-score归一化，得到归一化的数据矩阵。
4. 计算出100x100的相关矩阵，并可视化出混淆矩阵。（为避免歧义，这里“协相关矩阵”进一步细化更正为100x100的相关矩阵，100为学生样本数目，视实际情况而定）
5. 根据相关矩阵，找到距离每个样本最近的三个样本，得到100x3的矩阵（每一行为对应三个样本的ID）输出到txt文件中，以\t,\n间隔。

**二、作业环境（文件说明，函数说明，调用的函数库以及涉及哪些技术）**

### 实验环境

作业环境:Python 3.8

**实验导入的模块**

import pandas as pd

import numpy as np

from pandas import DataFrame as df

import matplotlib.pyplot as plt

import math

### 文件说明

AllDate.csv：存放实验一中的数据整理完成后的数据

实验二1-3：实验二的前三个代码信息

散点图.jpg：实验二中题目1的结果，可视化散点图

直方图.jpg：实验中题目2的结果，可视化直方图

zscore\_data.txt：实验二中题目3的矩阵结果

#### 数据的规范化

将体育成绩的四个等级bad、general、good、excellent转化为百分制对应的45、65、80、95的随机数，同理C6、C7、C8、C9由十分制转化成百分制

代码段:

for i in range(len(df)):

if df.Constitution[i] == "bad":

j=45

PE\_sum+=j

PE\_list.append(j)

elif df.Constitution[i] == "general":

j = 65

PE\_sum += j

PE\_list.append(j)

elif df.Constitution[i] == "good":

j = 80

PE\_sum += j

PE\_list.append(j)

elif df.Constitution[i] == "excellent":

j = 95

PE\_sum += j

PE\_list.append(j)

**三、难题与解决**

**实验二的1、2小题主要是呈现出可视化，所以相对比较快速完成，对于第3小题，之前没有想过要利用二位数组进行存储，参考了别人的代码才发现也可以这么做，按照他的思想去进行修改以及将每一门成绩进行Z-score归一化，得到归一化的数据矩阵并存储为zscore\_data.txt文件。**

**四、实验代码运行结果**

**# 1. 请以课程1成绩为x轴，体能成绩为y轴，画出散点图。**

**import pandas as pd**

**import numpy as np**

**from pandas import DataFrame as df**

**import matplotlib.pyplot as plt**

**import math**

**plt.rcParams['font.sans-serif'] = ['SimHei']**

**plt.rcParams['axes.unicode\_minus'] = False # 设置matplotlib正常显示中文和负号**

**#体育成绩用set方法进行量化**

**def set(pe):**

**for i in range(len(pe)):**

**if pe[i]=="excellent":**

**pe[i]=95**

**if pe[i]=="good":**

**pe[i]=80**

**if pe[i]=="general":**

**pe[i]=65**

**if pe[i]=="bad":**

**pe[i]=45**

**df=pd.read\_csv(r'E:\桌面\AllData.csv',encoding='gbk',header=0)**

**x=df.C1 #C1成绩**

**y=df.Constitution #体育成绩**

**C1\_list=list(x)**

**PE\_list=list(y)**

**set(PE\_list)**

**plt.scatter(C1\_list,PE\_list,c='blue',marker='x')**

**plt.title("课程1——体能成绩散点图")**

**plt.xlabel("C1")**

**plt.ylabel("Constitution")**

**plt.savefig('E:\桌面\散点图.png',bbox\_inches='tight')**

**plt.show()**

**#2、以5分为间隔，画出课程1的成绩直方图。**

**bins=[60,65,70,75,80,85,90,95,100] #设置间隔**

**plt.hist(x,edgecolor='blue',bins=bins)**

**plt.title('课程C1成绩直方图')**

**plt.grid(True)**

**plt.savefig('E:\桌面\直方图.png',dpi=300,bbox\_inches='tight')**

**plt.show()**

**#3. 对每门成绩进行z-score归一化，得到归一化的数据矩阵。**

**list=[] #存放所有成绩的列表，是二维列表**

**list.append(df.C1.tolist())**

**list.append(df.C2.tolist())**

**list.append(df.C3.tolist())**

**list.append(df.C4.tolist())**

**list.append(df.C5.tolist())**

**list.append(df.C6.tolist())**

**list.append(df.C7.tolist())**

**list.append(df.C8.tolist())**

**list.append(df.C9.tolist())**

**PE\_sum=0**

**PE\_list=[]**

**for i in range(len(df)):**

**if df.Constitution[i] == "bad":**

**j=45**

**PE\_sum+=j**

**PE\_list.append(j)**

**elif df.Constitution[i] == "general":**

**j = 65**

**PE\_sum += j**

**PE\_list.append(j)**

**elif df.Constitution[i] == "good":**

**j = 80**

**PE\_sum += j**

**PE\_list.append(j)**

**elif df.Constitution[i] == "excellent":**

**j = 95**

**PE\_sum += j**

**PE\_list.append(j)**

**PE\_average=PE\_sum/len(df) #体育成绩平均值**

**list.append(PE\_list)**

**PE\_SUM=0 #体育成绩-体育成绩平均值然后平方，最后所有项求和**

**for i in list[9]:**

**PE\_SUM+=(list[9][i]-PE\_average)\*\*2**

**PE\_var=np.sqrt(PE\_SUM/len(df)) #PE\_var表示体育成绩标准差**

**j=0**

**#每一列进行z-score归一化，j表示列数**

**while j<10:**

**if j<9:**

**average=df["C"+str(j+1)].mean()**

**var=df["C"+str(j+1)].std()**

**for i in range(len(df)):**

**list[j][i]=(list[j][i]-average)/var**

**else:**

**average=PE\_average**

**var=PE\_var**

**for i in range(len(df)):**

**list[j][i] = (list[j][i] - average) / var**

**j+=1**

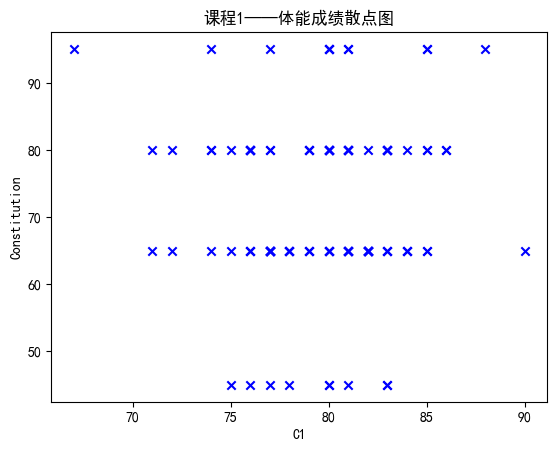
**data=np.mat(list) #用data形成list矩阵**

**data\_mat=data.T #data\_mat用来存data矩阵的转置，这样data\_mat就是归一化后的数据矩阵**

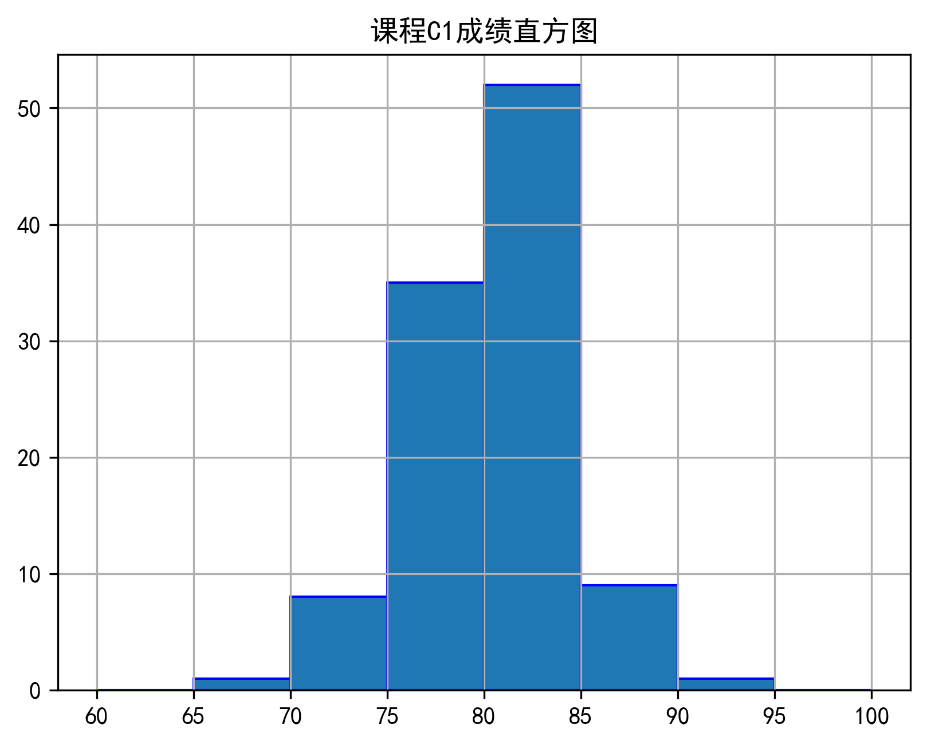
**# print(data\_mat[0])**

**df1=pd.DataFrame(data=data\_mat)**

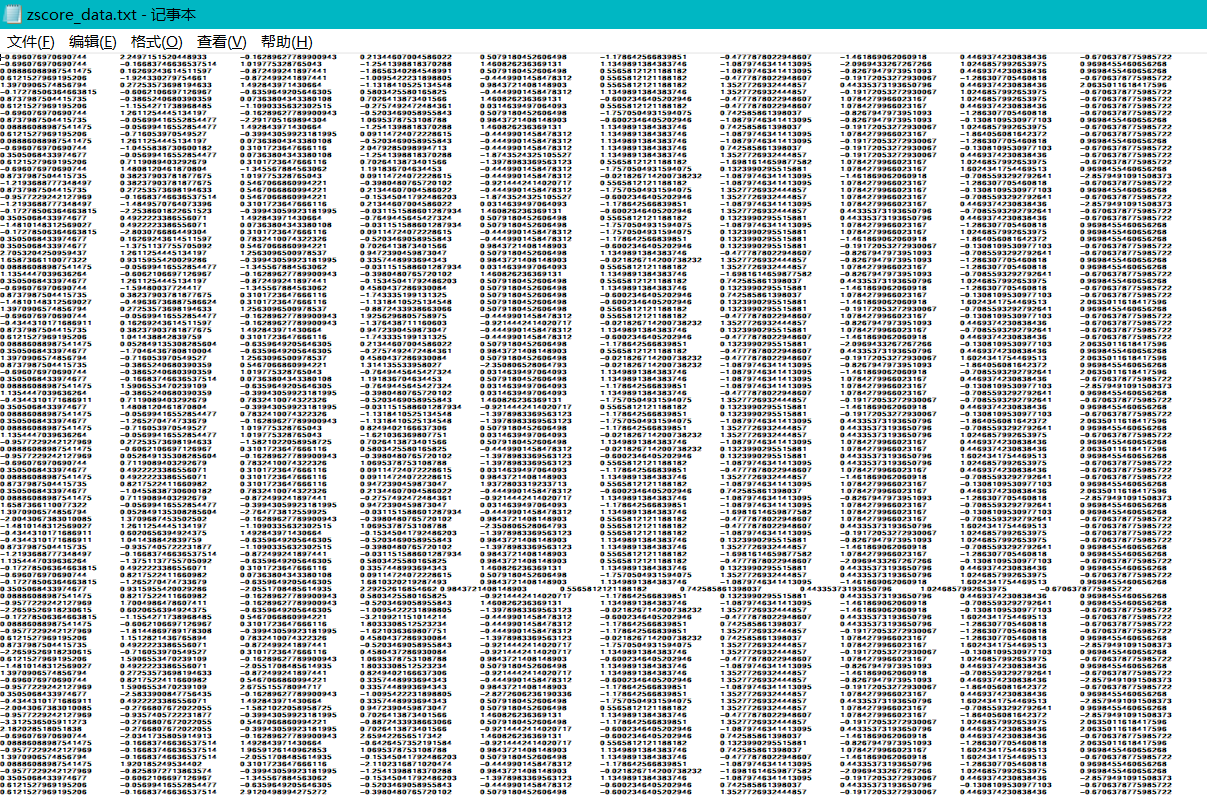
散点图



直方图



Zscore\_data.txt



**五、实验总结**

**相对其他同学来说，是比较晚做的，实现的部分也比较少，主要是可视化这一部分，相对来说比较简单，但对于python不熟悉以及对于在python中可视化一些模块也不是太清楚，读写文件也比较陌生，在网上查阅相关资料以及有借鉴于其他小组的github作业才得以完成，在这里再次致谢。**