暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | 软件实践AI算法B组 | | | | | | |
| 姓 名 | 游家伟 | 专业 | 吴健雄学院 | 班级 | 615182 | 学号 | 61518229 |
| 实验时间 | 2020.8.31-2020.9.23 | | 指导教师 | 孔祥龙 | | 成绩 |  |
| **一、实验背景和目的**  高考之后的志愿填报是每一位同学人生中最重要的事情之一，选择一个好的学校幸福终身，而选择了一个不合适的学校心烦意乱。在大数据时代，我们可以使用历年的学校招生分数作为参考，结合考生自己的兴趣，从而推荐比较适合该学生的学校以及专业。 | | | | | | | |
| **二、小组任务和个人任务**  我们小组的任务是实现学校以及专业的分类和分级。我个人承担的任务是针对每一个学校的每一个专业进行分级。 | | | | | | | |
| **三、个人任务需求分析**  我所需要做的任务就是把不同的专业，按照每一个专业的分数排名，按照不同的名次，将这些专业分成A,B,C,D四个大类。其中对于学专业的评级，对于每一年的综合而言，1-3为A，4-7为B，8-12为C，12以后都为D。也就是说通过对三年的排名做一个平均，得到最终的排名，之后再进行评级。但是由于数据比较乱，所以需要对数据进行整理，把同一个学校的每一年的分数先放在一起，之后通过sort函数进行排序，之后通过index函数返回排名，之后判断是哪一个等级。由于涉及到三个年份的同时运转，所以中间还是比较容易出错的。 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**    #对于学校的评级，对于每一年而言，1-3为A，4-7为B，8-12为C，12以后都为D  schid=[]#学校的id  major=[]#专业的名称  score=[]#专业的最低分数  year=[]#年份  province=[]  id\_2017\_m\_score=[]  id\_2018\_m\_score=[]  id\_2019\_m\_score=[]  import csv  with open('majorScore.csv','r') as f:  reader=csv.reader(f)  for row in reader:  schid.append(row[2])  major.append(row[5])  score.append(row[6])  year.append(row[4])  province.append(row[1])  for i in range(10002,10731):  for j in range(len(schid)):  if major[j]=='all' or major[j]=='All':  continue  if province[j]=='10':  if schid[j]=='%d'%i and year[j]=='2017':  m=[schid[j],year[j],major[j],score[j]]  id\_2017\_m\_score.append(m)  if schid[j]=='%d'%i and year[j]=='2018':  n=[schid[j],year[j],major[j],score[j]]  id\_2018\_m\_score.append(n)  if schid[j]=='%d'%i and year[j]=='2019':  k=[schid[j],year[j],major[j],score[j]]  id\_2019\_m\_score.append(k)  class level:  def \_\_init\_\_(self,schid,year,major,score):  self.schid = schid  self.year = year  self.major = major  self.score=score  def print(self):  print(self.schid,self.major,self.score)  ob\_2017=[]  ob\_2018=[]  ob\_2019=[]  for i in id\_2017\_m\_score:  ob\_2017.append(level(i[0],i[1],i[2],i[3]))  for i in id\_2018\_m\_score:  ob\_2018.append(level(i[0],i[1],i[2],i[3]))  for i in id\_2019\_m\_score:  ob\_2019.append(level(i[0],i[1],i[2],i[3]))  try: #通过lambda函数，只针对分数这一个属性进行排序  import operator  except ImportError:  cmpfun= lambda x: x.count # use a lambda if no operator module  else:  cmpfun= operator.attrgetter('score') # use operator since it's faster than lambda  ob\_2017.sort(key=cmpfun, reverse=True)  ob\_2018.sort(key=cmpfun, reverse=True)  ob\_2019.sort(key=cmpfun, reverse=True)  fin\_2017=[]  fin\_2018=[]  fin\_2019=[]  for i in range(10002,10731):  avr\_2017=[]  for j in ob\_2017:  if j.schid=='%d'%i:  m=[j.schid,j.major,j.score]  avr\_2017.append(m)  fin\_2017.append(avr\_2017)  for i in range(10002,10731):  avr\_2018=[]  for j in ob\_2018:  if j.schid=='%d'%i:  m=[j.schid,j.major,j.score]  avr\_2018.append(m)  fin\_2018.append(avr\_2018)  for i in range(10002,10731):  avr\_2019=[]  for j in ob\_2019:  if j.schid=='%d'%i:  m=[j.schid,j.major,j.score]  avr\_2019.append(m)  fin\_2019.append(avr\_2019)  ans=[]  for i in range(729):  for j in range(len(fin\_2017[i])):  q=fin\_2017[i][j][1]  if q in fin\_2018[i]:  m=fin\_2018[i].index(q)+1  else:  m=1  if q in fin\_2019[i]:  n=fin\_2019[i].index(q)+1  else:  n=1  a=(m+n+j)/3+1  if a in range(1,4):  ans.append([fin\_2017[i][j][0],q,'A'])  if a in range(4,8):  ans.append([fin\_2017[i][j][0],q,'B'])  if a in range(8,13):  ans.append([fin\_2017[i][j][0],q,'C'])  else:  ans.append([fin\_2017[i][j][0],q,'D'])  print(ans)  **通过代码，最后输出的是每一个学校的每一个专业的等级。** | | | | | | | |
| **五、实验结果与分析**  部分实验结果展示：  C:\Users\yjw\AppData\Roaming\Tencent\Users\1196993809\QQ\WinTemp\RichOle\T4`4CV7I_)XE_M[W5FV(`TG.png  可以看到，通过这一段代码，将三年的数据全部整合为一个等级，将各个学科很好的分成了A,B,C,D四个等级。同时，针对原本杂乱无章的学校排序进行重新编排，使得其按照学校的编号从小到大进行排序。但是还是有许多不足的地方。因为在评级的时候是对每一个学校都采取了相同的标准，前三为A。但是有些比较好的学校，例如东南大学，拥有远超3个的双一流学科，所以有一些好的学科会被划分为B类甚至是C类。所以，可能在对比两个整体实力有差距的学校时，具体学科的等级的参考价值并不是特别的大。 | | | | | | | | |
| **六、实验总结与心得体会**  本次实验最大的收获就是知道了数据收集以及处理的重要性。一个整理好而规范的数据对于后续的所有工作都有着很重要的作用。我在处理数据时，因为某一个大学的某一年数据没有，不得不对缺失的数据进行统一的处理。还有一些学校并没有公布自己的具体分数，只是用all代替了，所以我把这部分数据全部都舍去了。至于体会就是明白了团队合作的难度。一个人写一段代码实现功能可能并不是特别困难，但是要是与其他的组结合起来可能会变得非常困难，比如我们组在实现了所有功能之后，与UI组的同学对接还是产生了不少新的问题。未来，团队的合作开发肯定是主流的开发方式，所以培养这方面的能力是很有必要的。 | | | | | | | | |

2020年9月制