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 | 周之遥 | 专业 | 人工智能 | 班级 | 615184 | 学号 | 61518426 |
| 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**  高考志愿填报对考生来说事关重大，然而很多考生由于信息不够，面对自己的分数时，不知道该如何填报学校、专业。本项目旨在建立一个一体化的高考志愿填报助手，利用近几年的数据为考生推荐合适的学校以及专业。 | | | | | | | |
| **二、小组任务和个人任务**  小组的工作任务是通过聚类算法对分数、排名、高校进行分析，得到一系列分布模型，以助于后续的推荐。个人的任务是根据专业-一级学科的划分结果，在一级学科上对所有学校进行排名。 | | | | | | | |
| **三、个人任务需求分析**  首先需要对数据整理，把数据从学校-专业的形式变为一级学科-学校的形式。  在进行评级时需要采用了动态标准，因为一个一级学科所含学校数量（即拥有该一级学科的学校数）有多有少，不能简单地使用同一标准。  另外对于一级学科所含专业数不同、不同年份的问题，这里采用取最低录取线取平均的方式得到一级学科的评分。 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**  #读取数据并改变数据编排形式  csv\_file = "res.csv"  csv\_data = pd.read\_csv(csv\_file, low\_memory = False)#防止弹出警告  data = pd.DataFrame(csv\_data)  newdata={}  for collegeid in collegelist:  newdata[collegeid]={}  for i in range(len(data)):  collegeid=data.iloc[i]['collegeID\_id']  firstlevelid=data.iloc[i]['firstlevelIDs']  score=data.iloc[i]['minScore']  if firstlevelid!='None':  if newdata[collegeid].get(firstlevelid) is None:  newdata[collegeid][firstlevelid]={0:score}  else:  temp=newdata[collegeid][firstlevelid]  temp[len(temp)]=score  for collegeid in newdata.keys():  school=newdata[collegeid]  for firstlevelid in school.keys():  firstlevel=school[firstlevelid]  summ=0  for score in firstlevel.values():  summ+=score  rankscore=summ/len(firstlevel)  firstlevel['rankscore']=rankscore  for collegeid in newdata.keys():  school=newdata[collegeid]  for firstlevelid in school.keys():  firstlevel=school[firstlevelid]  print(collegeid,firstlevelid,firstlevel['rankscore'])  #按大类而不是学校编排数据  FirstLevelData={}  for collegeid in newdata.keys():  school=newdata[collegeid]  for firstlevelid in school.keys():  if FirstLevelData.get(firstlevelid)==None:  FirstLevelData[firstlevelid]={}  if FirstLevelData[firstlevelid].get(collegeid)!=None:  print('!!!')  FirstLevelData[firstlevelid][collegeid]=school[firstlevelid]['rankscore']  #排名  FirstLevelRank={}  for firstlevelid in FirstLevelData.keys():  dic=FirstLevelData[firstlevelid]  ranklist=sorted(dic.items(),key=lambda x:x[1],reverse=True)  rank=[x[0] for x in ranklist]  FirstLevelRank[firstlevelid]=rank  #评估函数  def GetThreshold(length):  a=int(0.05\*length)  b=int(0.2\*length)  c=int(0.5\*length)  d=length-1  return a,b,c,d  #进行评级  FirstLevelAss={}  for flid in FirstLevelRank.keys():  FirstLevelAss[flid]={}  rank=FirstLevelRank[flid]  if len(rank)<=4:  temp=['A','B','C','D']  for i in range(len(rank)):  FirstLevelAss[flid][rank[i]]=temp[i]  else:  a,b,c,d=GetThreshold(len(rank))  for i in range(0,a+1):  FirstLevelAss[flid][rank[i]]='A'  for i in range(a+1,b+1):  FirstLevelAss[flid][rank[i]]='B'  for i in range(b+1,c+1):  FirstLevelAss[flid][rank[i]]='C'  for i in range(c+1,d+1):  FirstLevelAss[flid][rank[i]]='D'  #改变数据编排形式并输出  FinalData=[]  for flid in FirstLevelAss.keys():  for shid in FirstLevelAss[flid].keys():  FinalData.append((flid,shid,FirstLevelAss[flid][shid]))  title=['firstlevelIDs','collegeID\_id','rank']  OutputData=pd.DataFrame(data=FinalData,columns=title)  OutputData.to\_csv("FirstLevelRank.csv", encoding="utf-8-sig") | | | | | | | |
| **五、实验结果与分析**    如图所示，201代表理论经济学这一个一级学科，前两名依次为复旦、清华。  最终的结果中水平较高的学校都在前列，但会有一部分大学排名虚高，这可能是因为在进行评估时使用了平均分，没有考虑到各省各专业的差异。 | | | | | | | | |
| **六、实验总结与心得体会**  在本次实验中，我们小组成员之间需要不断地沟通，处理好彼此之间接口的问题，达到最初的设想。这让我熟悉了作为一个团队如何去做好一个项目。  另外码云平台的使用相较于以往项目中完全依赖QQ群也有显著的优势，在代码的改动、版本管理等方面都方便很多。  在具体的实现过程中需要把握好目的，事先弄清一个基本的框架，然后再去写代码。 | | | | | | | | |

2020年9月制