暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | AI算法A组 | | | | | | |
| 姓 名 | 祁丁然 | 专业 | 人工智能专业 | 班级 | 091182 | 学号 | 09118209 |
| 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**  高考是走进大学校园的必经之路，是人生的重要路口，莘莘学子十年寒窗，为的就是可以在高考中春风得意金榜题名。高考后，学生面临的第一个重大选择就是选择大学，如何能够最大化利用自己的分数上到让自己和家人满意的好大学好专业成为高考后学生家长迫切需要解决的问题。  高考志愿填报助手应运而生，它能够帮助学生认清形势，摸清方向，根据学生分数和省份提供高考填报参考。我们组的任务为依据学生排名省份提供冲、稳、保三种大学志愿推荐。 | | | | | | | |
| **二、小组任务和个人任务**  小组任务：利用已有的大学录取分数和各个省份一分一段表，针对不同大学的录取情况做不同预处理，得到名次、省份、文理科类、大学的训练数据集，输入神经网络训练，得到的模型用于大学志愿预测，即志愿推荐。  个人任务：负责随机模拟第15-31所学校的数据，共37600+条，每条数据由名次、省份、文理类别、大学所组成，并对1-44所大学生成对训练数据集进行初步数据筛选，删除不合常理的数据，针对性重新生成数据。 | | | | | | | |
| 1. **个人任务需求分析**   上个学期，我们收集了88所大学的三年专业录取数据和各个省份的一分一段表，我负责第15-31所大学的数据集生成，根据学校、年份、省份、文理逐步筛查，找到大学省份年度科类录取情况，根据大学数据是否提供专业录取信息，选择不同的随机数据生成策略。  具体策略如下：  从学校、年份、省份、文理逐步筛查，查到该学校某年某省专业分数，取最低最高分，为了保证数据的正确性，即排除数据清洗不彻底或志愿滑档，对比最低最高分，差值高于50即认为数据失误，取次低分为最低分，由于专业分数为专业最低录取分，我们在最高分的基础上额外加了10分，作为该大学所有录取分数的最高分。对于像北方工业大学一样，没有专业分数的大学，我们将最低分额外加30分作为最高录取分数。又考虑到江苏省分数区隔小于其他省份，所以江苏省额外加分一律为10分。依据最高最低分，在一分一段找到对应排名人数，在这个区间内的所有人都有可能去该学校，取人数的1/18进入该学校，生成在分数段内的随机数，并换算查找为排名与学校、省份、文理共同生成一条数据。由于取加分的策略可能导致最高分溢出，即在一分一段表中无法找到数据，则将排名设为1。  此外，针对清北数据，额外根据招生人数生成清北数据。 | | | | | | | |
| 1. **实验过程（需附上关键代码及相关说明）**   **数据生成代码AI-statistic-qdr.py:**  读取数据：  college\_data = pd.read\_csv('大学信息.csv') # 读取大学数据  college\_batch\_data = college\_data[15:31] # 取15条数据  循环查找，查找到数据存储在temp\_data中：  **for** u in university:  **for** y in Year:  **for** p in Province:  **for** c in category:  temp\_data = csv\_data.loc[(csv\_data["College"] == u) & (csv\_data["Province"] == p) & (csv\_data["Year"] == y) & (csv\_data["category"] == c), ["College", "Year" , "Province", "category","score"]]  最高最低数据预录入：  **if** not temp\_data.empty: #存在dataframe  score\_list = temp\_data["score"].values.tolist()  score\_list.sort()  low\_score = score\_list[0] #录取最差专业的最低分，如xx的录取最低分  high\_score = score\_list[-1] #最高分  简单数据筛选，依据差值筛除不合理数据：  **for** ii in range(20):  **if** high\_score-low\_score > 50:  low\_score = score\_list[ii+1]  **else**:  **Break**  额外加分策略：  **if** high\_score != low\_score or p == "江苏":  high\_score = score\_list[-1] + addition  **else**: #没有专业区分，最低分与最高分相同，且避开江苏（江苏分段小）  high\_score = score\_list[-1] + 30  区间及人数生成，省略json文件读取，若存在json文件则存储为load\_dict：  **if** (str(low\_score) in load\_dict[str(y)][p][c]) & (str(high\_score) in load\_dict[str(y)][p][c]):  low\_rank = load\_dict[str(y)][p][c][str(low\_score)]  high\_rank = load\_dict[str(y)][p][c][str(high\_score)]  number = (low\_rank-high\_rank) // 18 + 1 #至少一名，整除18代表该分数段有18/88的学生去该大学  写入数据，并防止高分溢出：  **for** \_ in range(number):  random\_score = random.randint(low\_score, high\_score) #产生随机分数  **if** (str(random\_score) in load\_dict[str(y)][p][c]):  random\_rank = load\_dict[str(y)][p][c][str(random\_score)]  **else**:  random\_rank = 1 #说明random\_score超过了一分一段表的最高分  writer.writerow([random\_rank, p, c, u])  **清北数据生成代码 AI homework-清北数据 .py**  具体数据为手工录入，仅展示省份和清华文科数据：  Pro = ['湖南','广东','广西','海南','重庆','四川', '贵州','云南','西藏','陕西','甘肃','青海','宁夏','新疆', '北京','天津',  '河北','山西','内蒙古','辽宁','吉林','黑龙江','上海','江苏','浙江','安徽','福建','江西','山东','河南','湖北']  TW = [20,5,9,9,5,10,5,10,3,25,20,10,5,30,31,11,12,5,3,20,20,32,35,16,20,30,30,20,5,125,20]  仅展示清华(t)文科(w)代码（除上海sh浙江zj）：  **for** i in range(l):  c = "文科"  u = "清华大学"  **if** Pro[i] == '浙江' :  zj\_t+=TW[i]  **elif** Pro[i] == '上海':  sh\_t+=TW[i]  **else**:  **for** \_ in range(TW[i]):  random\_rank = random.randint(1, TW[i])  #random\_rank= TW[i] - random\_score % TW[i] +1  writer.writerow([random\_rank, Pro[i], c, u])  不分文理省份数据北京大学(p)生成代码（上海）：  **for** \_ in range(sh\_p):  num = sh\_p  c = "all"  u = "北京大学"  random\_rank = random.randint(1, num)  #random\_rank= PL[i] - random\_score % PL[i] +1  writer.writerow([random\_rank, '上海', c, u]) | | | | | | | |
| 1. **实验结果与分析**   普通数据生成：  rank province category college  376 北京 文科 中国传媒大学  446 北京 文科 中国传媒大学  576 北京 文科 中国传媒大学  280 北京 文科 中国传媒大学  ......  3098 湖北 文科 华东理工大学  1391 湖北 文科 华东理工大学  2188 湖北 文科 华东理工大学  2282 湖北 文科 华东理工大学  2100 湖北 文科 华东理工大学  清北数据生成：  rank province category college  19 湖南 文科 清华大学  4 湖南 文科 清华大学  20 湖南 文科 清华大学  8 湖南 文科 清华大学  17 湖南 文科 清华大学  ......  141 浙江 all 北京大学  6 浙江 all 北京大学  3 浙江 all 北京大学  171 浙江 all 北京大学  结果分析：  实验数据集输出为csv，列分别代表排名（rank）、省份（province）、科类（category）和大学（college），该数据集省略年份，方便神经网络训练，也扩大同类型数据的数据量。  我负责的15-31所大学生成的数据集共376850条数据，根据由招生人数生成的清北人数，生成的清北数据集共6109条数据。  该数据集能够排除数据清洗不完全和一分一段json文件缺失导致的错误。但是由于json文件缺失，数据量并没有达到预期。由于大学编码顺序为省份，区间人数取1/18只是大略近似，如果采用某一综合指标排名编码大学，可以采用更精细化的分段人数去向选择，更进一步，如果可以得到各个大学各省招生人数，则数据集的精确度将大大提高。 | | | | | | | |
| **六、实验总结与心得体会**  本次高考志愿填报助手项目，我们AI算法A组主要负责使用神经网络推荐大学。我负责部分数据集的生成，将15-31所大学的数据提取筛选为神经网络需要的数据，并利用组内成员提供对清北招生人数数据生成清北数据集。  当我们组确定了我们的目标后，我们就陷入了迷茫之中，我们组应该选择什么样模型去完成我们的目标，我们的模型的输出是什么，我们的推荐策略是什么。经过常常长长的小组讨论，我们最终确定了使用神经网络来训练数据，也确定了数据集包含什么类型的数据。  最终根据我们的目标，我们几个人完成了数据集的生成，我们的数据集排除了数据清洗不完全和一分一段json文件缺失导致的错误，也根据不同的大学有选择地取人数区间，但是由于一分一段json文件缺失，导致数据集的量不够，大学数据的省份划分也导致数据集的准确度稍有下降。  本次实验不仅锻炼了我编写代码，更让我学到了如何在团队中工作。在我们聚在一起开小组讨论会中，我学到了如何推进项目，如何进行任务分析，如何将自己和别人的想法结合，如何将我们的想法付诸实践。小组与小组之间的合作是整个项目能够顺利出成果的基础，在小组和小组对接中，我学到了整个项目是如何开展的推进的。在组内工作也让我收益良多，小组工作既要分工明确，工作量适中，也要通力合作，共同努力，这样才能有序完成任务。  从迷茫到找到方向，从无从下手到深夜改代码，这个项目让我收获最大的是负责，由于我们的AI算法的最大依赖就是数据，数据就是算法的基础也是决定算法能不能有好效果的最大影响因素，所以我觉得我们负责数据集生成的这些组员身上的责任很大，所以对每一个可能的影响数据集的因素我们都不放过，希望能够利用已有的数据，得到最好最准确的数据集。我也认为我已经成功地将我们能够想到能够实现的数据集生成策略全部实现。  由于本次项目小组过多，我没有参与到其他的组的工作，没办法学到整个项目完整的开发过程，还是有些遗憾。希望有机会可以全过程的参与项目的开发，积累经验教训，为以后的学习工作打下坚实的基础。 | | | | | | | |

2020年9月制