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 | 王昕彤 | 专业 | 人工智能 | 班级 | 091182 | 学号 | 09118205 |
| 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**  新冠之下，为弘扬社会主义人文关怀，本次实践希望设计一个基于Web的高考志愿推荐系统，为即将进入大学的莘莘学子提供引导和帮助。  高考志愿推荐系统意在帮助高考完的考生根据自己的分数和所在省份的情况预测自身适合的大学与专业，有助于考生更加了解自己、明确目标定位。上学期，我们收集了各个省份的一分一段表以及部分高校的专业录取分数线，短学期内，我们被分成各个目标不同的小组，分组完成包括数据获取及清洗、知识图谱构建、算法模型设计、UI界面设计、Django开发等任务，最终实现高考志愿推荐系统的初步构建，并长期进行维护与更新。 | | | | | | | |
| **二、小组任务和个人任务**  本组计划用AI算法实现系统的学校推荐功能，根据学生的高考分数及排名，以及所采集大学的录取分数线和录取人数，训练一个能够预测学生报考时冲稳保的学校的神经网络分类器。  根据计划分析，小组的主要任务共分为数据收集、数据编码处理、数据模拟、数据清洗、神经网络模型构建、接口设计。其中，本人的任务主要为进行模拟数据的生成。其中，每条数据由学生名次、学生所在省份、学生的文理类别和学生被录取的大学组成。  具体来说，我需要根据各个省份的一分一段表以及88所高校的专业录取分数线，随机模拟若干名考生成绩（名次）、省份及所选大学作为预训练数据。模拟数据作为神经网络的预训练数据，模拟得越真实才能够训练出符合实际情况具有现实意义的模型，因此，此任务除了代码还需要人工进行检查处理，才能够保证数据尽量有效。 | | | | | | | |
| 1. **个人任务需求分析**   训练集所需要的数据包括学生名次、学生省份、学生的文理类别和学生被录取的大学，由于学生的高考分数及排名数据是不公开的，我们无法获取，所以首先我们需要通过往年的高考录取情况进行数据模拟，生成训练需要的数据。而通过往年的学校录取分数线和一分一段表可以大致估记出学校在各省份录取的学生的排名区间，再根据此排名区间随机生成范围内合理数量的名次数据，最终整合成所需数据。  因此，要完成数据模拟的任务，需要的数据为各个省份的一分一段表和高校的专业录取分数线，而这些数据在春季学习已经基本收集完毕，但由于部分错误及缺失，需要进行清洗及更进一步的爬虫工作得到完整的数据。 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**  首先考虑大多数有录取分数线的高校及其有一分一段表的对应省份。  对于这种情况，可以通过代码实现数据模拟。大致思路如下：对于某一年某高校在某省份的录取分数，最低分即为录取分数线，最高分设定为最低分加addition分数，其中addition为超参数，可以通过尝试确定合适的数值，通过查找定位，将最高分和最低分对应的此省份当年最小名次和最大名次确定下来，得到此高校在此省份的录取名次的范围，我们假设这个分数段内共有n所可以选择的学校，那么我们将随机生成总名次数除以n条排名，此大学及省份就是对应的大学及省份。  首先将年份、学校、文理科、省份存为字典，然后读取csv（包括各省份一分一段表和大学录取分数线）。代码如下：  Year = [2017,2018,2019] Province = [**'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'**,**'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'**,**'香港'**,**'澳门'**,**'台湾'**] category = [**"文科"**,**"理科"**, **"all"**]  result = open(**"09118205王昕彤.csv"**,**"w"**,encoding=**"utf-8-sig"**,newline=**''**) writer=csv.writer(result) writer.writerow([**"rank"**,**"province"**,**"category"**,**"college"**])  通过循环，找到匹配的分数线和排名，判定是否为特殊情况，若是，则进行特殊处理，将排名放缩到合适的范围。若不是，则直接通过分数线对应的排名范围进行随机生成名次和对应学校省份，将其存储在另一个以我学号姓名为名字的csv文件内，此处有超参数addition：  **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"**]]   **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* **for** ii **in** range(20):  **if** high\_score-low\_score > 50:  low\_score = score\_list[ii+1]  **else**:  **break  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  **if** c == **"不分文理"**:  temp = **"json/"** + str(y) + p + **"all"** + **".json"   else**:  temp = **"json/"** + str(y) + p + c + **".json"    if** os.path.exists(temp):  **with** open(temp,**'r'**,encoding=**'UTF-8'**) **as** load\_f:  load\_dict = json.load(load\_f)  **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) // 25 + 1 *#至少一名，整除18代表近似该分数段有1/25的学生去该大学* **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])  根据具体学校情况尝试超参数，运行代码，完成第一部分。最终addition数值为10：addition = 10  对于需要人工处理的学校，因为如清华北大等尖端大学无法用简单的逻辑进行模拟，于是需要自己查找各省份清华北大录取人数及分数线，然后通过模拟生成。 | | | | | | | |
| **五、实验结果与分析**    最终生成的数据每一条为一行，包含排名、省份、文理科、学校。  通过最终将生成的数据作为训练集对模型进行训练后得到的结果，发现数据的质量仍然不理想，  经过分析，主要的原因是对于不同的大学及省份，有着不同的录取政策和情况，所以无法一概而论，用同一种参数进行处理，但因为时间的限制，我们只能去假设这些大学的情况近似相同，然后去尝试出一个最合适的参数，所以最终得出来的数据质量不稳定。而对于清华北大这种顶尖高校，首先因为一分一段表的高分部分无法爬取，所以我们无法用代码进行处理，所以只能够人工通过经验进行分析，同时不考虑年份，最终手动完成了这些大学的数据生成。总的来说，实验得到了初步的完成，有了一个可以接受的结果，后续的维护中我们可以更仔细地对各个大学都进行单独的数据生成，用新的数据进行训练，从而更新模型，达到更好的效果。 | | | | | | | | |
| **六、实验总结与心得体会**  实验总结：在此次实验中，我们小组将之前学习的专业知识灵活运用，在一次次的尝试中最终  成功得到了一个初步的神经网络，完成了输入成绩及排名得到三种推荐学校的目标。而我个人则完成了准备预测数据的任务。虽然在过程中遇到了一些问题，有些甚至还没有被解决，但在时间限制下，我们准备预测数据的小分队经过了很多激烈的讨论，仍然达到了有限条件内最好的结果。  心得体会：总得来说，在这次实验中，我收获良多。首先，能够将所学得以应用，使我获得巨大的成就感，让我觉得自己学习都是有意义的。其次，小组合作是一项充满挑战的工作方式，在此次实验的过程中，在遇到困难时，队友相互讨论可以提高解决问题的效率，但和其他小组的联系不够紧密，导致数据需求无法被迅速满足，从而延迟进度。最后，在过程中，我察觉到自己很多的不足，比如效率低，以及注释不够清晰导致其他同学很难直接看懂。未来，在维护的过程中，希望能够不断完善自己，帮助团队不断完善整个项目。 | | | | | | | | |

2020年9月制