暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | AI算法应用1组 | | | | | | |
| 姓 名 | 邵彤 | 专业 | 人工智能 | 班级 | 091181 | 学号 | 09118145 |
| 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**  我们暑期学校正值高考出分阶段，结合我们自身在当时填报志愿的迷茫焦虑等等情况，我们认为有必要为高考学生们开发出一套可以通过考生自己的省份以及高考分数尽可能准确的预测其能够填报的大学以及相关专业；同时我们还要考虑到个人对特定专业，特定地域等等外部因素的影响，对最基本的大学以及专业预测的功能进行扩展，最终能够根据考生的分数排名，省份以及爱好等等因素综合考虑来进行预测；同时，充分利用往年的数据，可以建立简易的知识图谱，以分析各种数据的特点并从中进行一些操作等等；此外，我们将整个项目以网页形式展现，能够实际使用，综合利用后端的数据库，AI算法模型，知识图谱，以及前端的网页显示与功能展示，将我们的整个项目形成一个完整的，可使用的系统，并考虑挂在学校服务器进行部分的开放测试。 | | | | | | | |
| **二、小组任务和个人任务**  本小组为AI算法1组，我们小组的主要任务是利用前三年考生的高考数据，包括文理科，分数，排名，省份等等信息，建立AI算法模型，对现有考生可能考上的大学进行预测，并且在最终根据计算得出概率的大小，简单分为三个梯度：冲一冲，稳一稳，保一保，每个梯度3所学校共9所学校。我们经过简要的数据与成果分析确定使用神经网络，令输入为每一位学生的分数排名百分比，大学位置信息，经过神经网络的训练，最终可以按照概率大小生成上述大学列表，并且为了外部因素尽可能不影响模型的生成效果，我们将考生个人的兴趣等等倾向因素全部放在模型之外以权重的形式添加，这样既可以有效的将此信息放入模型的预测结果同时尽可能避免了个人特例作为噪声对普适模型的影响。  而我个人的任务是将数据准备组给予我们小组的数据进行维度上的放缩，满足我们模型的数据格式，同时符合本小组模型的维度需求，同时根据大学的ID，省份，分科以及分数，在五分一段表的基础上对各个分数段上下浮动生成多组数据以满足模型对庞大的数据量的需求。 | | | | | | | |
| **三、个人任务需求分析**  我的任务主要有三个：整理训练数据的格式；缩放原数据的维度；扩充数据量。  首先，整理训练数据格式，我需要将原有的数据集(即省份分科分数json文件以及总的录取分数线csv文件)进行合并，形成考生录取分数，分科情况，考生省份，录取大学的一张数据表，而根据我们的分析，由于每一个省份分数之间存在较大差异(如江苏的总分为480)，所以我们先需要根据考生在其省份中的分数，随后寻找其在整个省份中的排名(根据五分一段表)，将分数数据转化为排名数据，在这操作过程中注意点：第一个是原有数据的名称不对应问题，有些文件中对不分科命名为“不分文理”，而有些则为“all”，这其中带来的错误需要避免(若忽略可能导致数据读取不全)；第二点则是建立纠错步骤，原数据文件有个别存在符号缺漏或是冗余等现象，由于数据量较大，手动检查较为困难，所以在读取时一定需要进行检测纠错，主动的将代码错误与原文件格式错误区分开，避免在debug时花费不必要的精力在这方面。  其次，扩张原数据的维度，经过整理过的数据为排名，省份，分科，大学4个维度，此为基础数据文件，之后会再进行二次的维度添加，例如大学位置的经纬度信息等等，所以这里相当于所有具有复杂关系的录取数据缩减到现在的4维数据，方便之后的维度二次添加，以及训练。  最后，扩充数据量，我们的模型对于数据量的要求较大，所以我们需要现有的数据合理的先预测生成出更多的数据，才能让我们的模型学习到比较有效的特征。这里主要根据已有五分一段表进行这一操作，五分一段表显示了同一分数所含的人数，我可以根据相邻两分数的差值(即该分所含人数)，但是由于其并不只是一分内包含的人数，而是一个小区间(五分)，所以我们只能够大致估计出可以扩张多少数据，并且为了保证数据的随机性，我们并不能直接将该分数段数据乘上其扩张的倍数，这样会导致大量的数据冗余而使得大部分训练数据没有意义，所以我们需要为扩张数据添加一个浮动值，以保证训练数据的不一致性，最终经过浮动值与扩张数的不断调整，可以得到数量可观且数据特征仍然与原数据近似的模拟数据集，以用于我们后面的模型训练。 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**  首先，通过高校录取分数线整合文件读取到我们所需的信息并转化为表格：  csv\_data = pd.read\_csv('高校录取分数线整合（省份名字统一）.csv')  college\_data = pd.read\_csv('大学信息.csv')  college\_batch\_data = college\_data[58:73]  university = college\_batch\_data['collegeName'].values.tolist()  universityID = college\_batch\_data['collegeID'].values.tolist()  其次，设置分数的浮动分值：  addition = 15  设置大学，年份，省份，分科等等变量  然后以上面四个变量建立四重的嵌套循环：  for i in university:  for j in Year:  for k in Province:  for l in category:  ……  以此遍历文件所有数据，首先获取数据：  temp\_data = csv\_data.loc[(csv\_data["College"] == i)  & (csv\_data["Province"] == k) &(csv\_data["Year"] == j)  & (csv\_data["category"] == l), ["College", "Year" , "Province", "category","score"]]  由于可能会有错误数据存在，所以我们并不能按行读取并直接赋值使用，而是需要根据设置进行判断，忽略掉错误数据，但是这样也可能忽略掉一些必要的数据，所以我们在遇到错误数据时让循环发出提醒，之后再在原文件中手动将错误数据修改正确。  再然后，利用上述的数据形成列表，进行排序，可以得到某大学的录取最低分与最高分，这里注意的是由于在添加浮动时同意规定向上浮动(否则相邻分数段之间的分数可能会被执行两次分数浮动操作)：  if not temp\_data.empty:  score\_list = temp\_data["score"].values.tolist()  score\_list.sort()  low\_score = score\_list[0]  high\_score = score\_list[-1] + addition  之后，打开各省份在各年份下的五分一段表，若上述最高分最低分在表范围内，则将分数直接转化为排名：  if (str(low\_score) in load\_dict[str(y)][k][l]) & (str(high\_score) in load\_dict[str(y)][k][l]):  low\_rank = load\_dict[str(y)][k][l][str(low\_score)]  high\_rank = load\_dict[str(y)][k][l][str(high\_score)]  设置近似人数：k=30  并且以此循环，利用之前最高分，最低分的区间生成随机分数(即相当于向上浮动)，在用五分一段表将模拟分数转化为模拟排名：  random\_score = random.randint(low\_score, high\_score)  if (str(random\_score) in load\_dict[str(y)][p][c]):  random\_rank = load\_dict[str(y)][p][c][str(random\_score)]  同时我们还需要考虑到不在五分一段表内的情况，这部分的数据由于没有每个分数段的人数数据支撑，所以我们无法得出具体的扩张策略，后面我们在这部分中采用手动添加数据的方法，同时对于某些极端分数段不做改动，因为此类数据多呈正态分布，若盲目扩张其两端的极端数据，极有可能造成多峰的出现从而破坏了原有数据的分布特征，所以对于该类数据我们基本保留录入。在代码层面，我们直接设置不在五分一段表内的数据的排名为1：  random\_rank = 1  以此作为标记，方便后续的手动添加。 | | | | | | | |
| **五、实验结果与分析**  最终，经过浮动分数和近似人数设置的不断调整，我们按照每15所大学为一个区间，基本可以保证每一个区间内生成大约28-36万左右的录取数据，可以满足我们模型对于数据量的需求，而原数据中大约有8000条数据不符合筛选条件，所以在此基础上我们手动添加了这些数据，并且都没有文理科等等因素的影响，完全按照分数段包含人数添加，所以我们可以认为这样的数据是可用，有意义的。  同时，在添加过程中，我们意外发现json文件与csv文件中的分科命名不一致，由于之前的防错误筛选，所以导致了所有不分文理的省份(主要是浙江，上海)的考生数据丢失，之后我们在修改错误后重新得出了正确数据，但是也能够提醒我们代码方面的不足，所以之后我们为此也添加了未读取提示。  最终，经过多次的修改调整，我们可以为我们的神经网络模型提供充分而又符合实际意义的考生数据，具体格式为：    最终本段数据量总数为334500。 | | | | | | | |
| **六、实验总结与心得体会**  这是我们第一次经历如此大的项目，小组多，小组人数也多，在组间组内的配合和协调我认为是我们学到的最多的东西。包括像码云这样的大型项目合作平台，虽然不是第一次用，但是真正利用它进行项目的推进也是第一次，我们与各组的协调尤其是与数据准备组的对接配合以及与网页展示组的协商等等，让我们对于类似这种庞大的项目工程有了一些初步的认识。并且在代码方面也能够有所体现，主要就是在接口方面，为了方便前后工作的展开，代码基本上能以函数或者是外部文件引入的形式在其他地方使用，像类似于文件使用相对路径而非绝对路径等等方面都需要有注意，让我们对这样类型的大工程有了一定的基础。  同时，对于我们小组来说，主要工作是利用深度学习模型根据考生数据去预测报考的大学，这也是我们将学习的理论模型应用于实践的一次宝贵尝试，实际情况包含数据等等情况肯定不会完全如我们预期一般完美，因此实际问题到底适合什么样的模型，针对具体情况需要哪些细节上的微调等等都是我们需要考虑的，但是最终我们的模型有了不错的效果，而这种将理论模型在实际问题上的应用也让我们对于模型的迁移，微调等等方面以及可能产生的一些后果有了更加清楚的认识。  而对于我个人而言，更多的则是个人在团队中位置的清晰认识，像如此庞大的工程，即使个人能力再强，也不可能完成所有，即使能完成，时间条件也不允许，而这就需要同伴之间的配合；此外，让别人明白自己的想法非常重要，尽管有时的新颖思想会让我们的工作或者是功能看上去非常出彩，但由于所有人的工作是并行展开的，若一个人临时变化，势必会与他人的工作形成冲突，导致不必要的时间精力浪费，所以在出现变化或者想法时，要第一时间与团队沟通，让他人明白自己的想法，才能根据别人的想法对自己的工作进行微调，或是让别人按照自己的想法增删功能等等。  在这次项目中，虽然可能个人工作量并没有预想中的那样繁重，但是这样的项目团队模式，还是让我学到不少，更是对大型团队合作，模型迁移与实际问题的应用等等方面有了进一步的认识，对我们有着极大的好处。 | | | | | | | |

2020年9月制