廣東工業大學

QG 中期考核详细报告书

题　　目　数据挖掘（Pass_Or_Not）

学　　院　计算机学院

专　　业　计算机类

年级班别　20 级（15）班

学　　号　3120005313

学生姓名　詹培林

2021 年　4 月 17 日

Pass Or Not

詹培林*　　导师：张平路

（计算机类，计算机学院）

摘　要：　本文主要介绍在进行中期考核时进行数据挖掘的详细过程：包括过程中必要知识的学习，以及进行时的数据清洗、分析，特征工程，模型选择、评估的内容。

关键词：　数据分析

1 数据集的查看

1.1 查看数据集的变量的说明

不难从直观感受有几个特征会比较大地影响学员的表现和影响受训结果，正如：测试难度级别、性别、教育水平、居住城市的登等级、年龄、是否残疾。最终是要预测出是否通过，从这可以认出是二分类问题（非零即一）。有了大概的信息印象，接下来进行数据集的细节方面的了解。

1.2 查看数据集详细内容

首先，需要导入数据到 Jupyter 中，我根据师兄的建议建立了三个文件(后来其实非常多，由于比较时而卡顿，担心修改原有的代码被误删还找不回来)：数据进行探索性分析的文件，特征工程的文件，以及模型训练的文件。

首先，导入了一些必要的库，例如：正则表达式使用的’re’库，针对数据操作的’numpy’,’pandas’以及模仿师兄导入了使得结果可以同时显示的’InteractiveShell’模块(能够让其结果显示在同一个行中)。

接下来进行：数据集的各类信息查看。先利用 pandas.read_csv() 方法将数据导入，用.head(10)查看前十行的基本情况发现：特征类型不仅有数字，还有英文字符，而且特征比较多。

再利用.describe()和.info()的方法，查看数据的数量等各种信息：这里主要关注到年龄：是否有异常值（即非正常人的年龄数值），大小值、均值（代表了这群人的普遍年龄水平，可以考虑年龄相近，应该不是很大的影响因素），接着依次对各类数值进行思考（与刚看到数据集变量说明的时候进行对比）。不过还是应该全面的查看，直觉不能代表一切，数值才能说明事实。

2 数据集的处理

2.1 缺失值处理

在查看数值的时候很容易发现年龄缺失值非常多，其他数值缺失较少。我尝试把在年龄特征所缺失的空上补平均值，并把其他 NaN 所在的行都删除，来看看是否会缺失很多，结果证明损失了数据还是将近 9000 左右，这与样本总数占比还是挺大的，我决定放弃这个做法。因为，其他数值确实不多，查看各属性出现频率后，我决定利用随机填补的方式进行填补（在单独每列，取邻近的上边或者下边的值去填充）。猜想原因：如果一个数据比较多，那他的分布优势比较大，而缺失值少，填充上去后，结果影响不大；反之数值少的也很少有机会被复制。这应该是比较好的方案。

而有些值，例如 id 号是与唯一 id 有紧密关系，其实就是一前一后的关系(例如:1464_123, 前后分别 trainee_id 和 test_id），为了能够真实反映，我也使用了正则表达式去匹配对应数值，然后加以修改。

因为只把认为重要的缺失值进行填补，思考着可能还会有缺失值，最后在进行清除剩余含有缺失值的行。

2.2 针对不同特征进行不同方式编码

由于每个特征中的属性存在多种或只存在两种，进行编码方式应该不同。

例如：针对 test_type、gender、handicapped 等属性各自特征内只有两种属性，可以进行 0,1 编码，表示彼此的类型。而对于 difficult_level,education 等数据要进行能够显示等级的编码，从小到大可以是 0,1,2,3...因为不同等级之间是有不同距离的，这可以从直观上感受，比如说：十分容易，容易，适中，困难，十分困难的等级区别，任意两个等级的距离不一定相等的。

而如果已经是存在相应等级的数值，则不需要去进行编码，例如 city_tier

而对于程序类型（program_type）有七种类型，这里我不清楚各类型之间是否有等级区别，应该在困难等级那里体现了，所以这里进行独热编码。为此，在网上浏览了 csdn，查找了 preprocessing.OneHotEncoder()的使用方式。编码后将这每样本 7 个 0 或 1 当成一个新的特征取考虑，49000 左右样本数，每个样本 7 个二进制数似乎会会使得内存不足，在第 48600 左右后面出现的新特征全变成了 NaN。我也因此忍痛.dropna()。（后来发现相关性极低，因此删除了，也没什么变化）最后再进行暂时的.to_csv()转存。

2.3* 改变思路：拟合年龄（补充）

由于，多次下方工作尝试提交之后，发现正确率并没有很高，回想到辜师兄在群内提及可以试着拟合年龄。我因此思考着或许缺失的年龄实在太多了，大概率会影响整个人群的平均年龄，确实不应该直接用原数据集的平均年龄去填补。但如何拟合年龄呢？

思路：这或许也是拟合目标值一样的方式：只不过是把年龄当做目标值。其他（包括原本的目标值）当做特征。由此，我又创建了 EDA2 文件进行拟合。

过程：同上方准备工作大致相同，随机填充后（这里不进行 dropna，为了后来得到的年龄数据能够一对一的填补），把感觉[1]可能和年龄有相关性的特征值(参与度,难度水平,年龄,教育水平,城市等级,残疾与否,测试类型,是否通过)当做测试集中的特征，利用 LinearRegression 进行训练和预测，这里没有进行评估。最后，逐个遍历年龄值，如果判断出是为空[2]的，就使用点对点的填补。

[1] 如何感觉：根据年龄反观其他特征：不同年龄在社会有不同的角色，不同角色造成的角色特征是不一样的，简单且一般来说，一个妇女，在一定年龄的时候会抚养刚出生不久的孩子（也有其他方式抚养），因为空闲时间不足，她就可能就在培训中的参与度不足，使得参与度比较低，通过率可能也降低。当然，这并不是一一映射的，也有其他可能。同样呢，我们也可以思考着，年龄不同的人选择的测试的难度级别、测试类型也都是会不一样的。在这，也只是简单的提出一些不大成熟的联想。

[2]进行判断时候是否为空的时候遇到了比较难处理的问题，我分别利用判断是否为’[空格]’,’[什么都不填]’，’[nan]’,’[NaN]’...都无法识别。而返回到之前的查看数据 info()，发现他不是 np.nan 类型而是 np.float64,这要怎么判断呢。进行经过查询 csdn 了之后，可以用一个很特别，很奇怪的方法，使用 i != i，来判断。

3 特征工程

3.1 选取特征

3.1.1 理解业务

要判断学员是否能够通过认证培训，与其相关的外在因素应该会有所选的类型、难度、和所处代理的城市的等级、测试的类型，而内在因素应可能是学员的年龄、教育水平、身体因素与参与度因素。因此，首次利用线性回归尝试去完成分类问题时，选取的就是以上特征。由于结果不尽人意(图 3.1 所示)。准确率低至师兄们说的全填‘1’都能近 70 的正确率。

```
总测试数: 49998
判断正确的数量:  34813
正确率:69.62878515140606%
```

图 3.1

因此进行下方操作。

3.1.2 特征与目标值的相关性

基于方差分析，检验 f 值，利用 SelectKBest 选取特征，利用柱状图进行显示占优势的特征。再来选取"difficulty_level", "city_tier", "age","test_type"4 个优势特征。(图 3.2 所示)

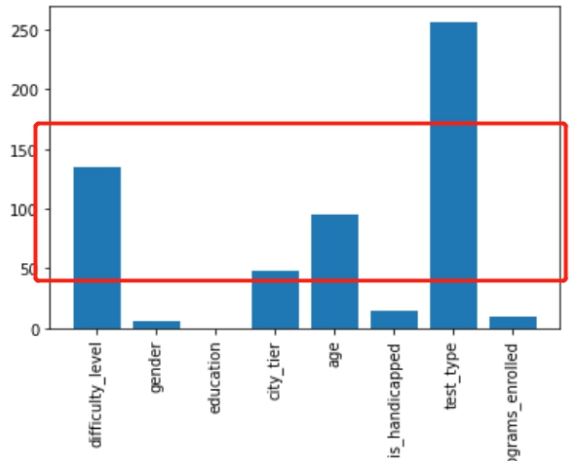

图 3.2

3.2* 选取特征（补充）

其实这个过程尝试过很多次，一直达不到很好的结果。似乎失去点什么，在后来重新编码、重新选取的时候，是把程序类型的特征重新放了回去，这次对此特征的编码不再是独热编码，而是简单的 0 到 7 的编码，因为重新思考后认为：属性 S 到 Z 其实已经显示之间的顺序了，也就是距离是不同的(当然，或许是一个巧合)，结果是它也有较大的相关性(图 3.3 所示)。

我又因此将这个特征选入，重新进行模型训练。

但是它似乎对准确率没有什么影响。为此还仍然在怀疑有其他不起眼的特征还在作用着。

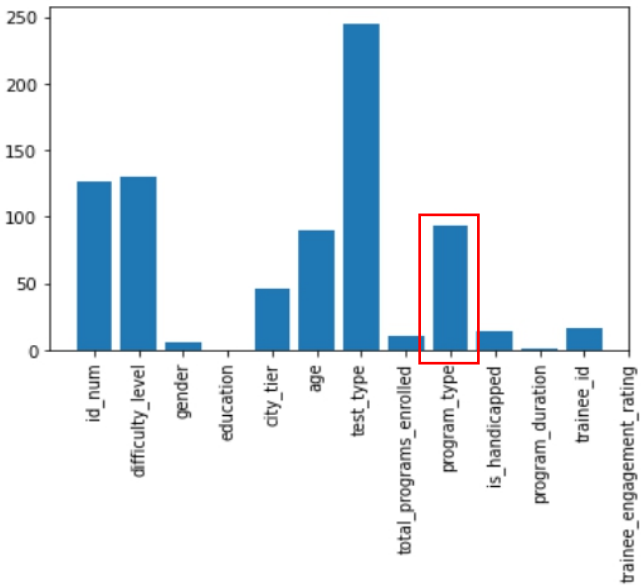

图 3.3

4 模型训练与评估

将所选的特征合成新的特征组与目标值放入 LinearRegression 训练，又将所有训练集当做测试集去预测。由于这种线性回归得算法所得结果不会像分类算法那样将结果整整返回 0 或 1，则将结果进行简单的处理：接近 1 的(>=0.5)赋值为 1，靠近 0 的(<0.5)赋值为 0，将所得的结果和原本的训练集的目标值进行对比，使用 predicts == ini_data['is_pass'] 进行 bool 运算，在利用聚合函数进行求和 sum(predicts == ini_data['is_pass'])。将所得的值和总预测数进行求商，也就是得到准确率，接下来再使用 confusion_matrix 和 recall_score 对该模型进行评估(图 4.1 所示)。

由下图可知，准确率并不高，召回率高得过分，其实可以理解为就是基本找不出无法通过的学员，模型效果一般，还把绝大部分的学员都评判为通过。这个原因一直未弄明白。

```
准确率:69.74077246046832%
混淆矩阵：
 [[    67  14435]
 [    51  33320]]
召回率：  0.9984717269485481
```

图 4.1

5* 感想

在过程中尝试过很多方式，从刚开始模仿师兄线性回归、逻辑回归、随机森林（集成算法）等算法，结果并不如人意，加上不清楚其中他们的原理，处于 70%以下的准确率也不足为奇。但明白的是，这是由于在数据集处理方面和特征工程方面的火候还不够，没有前面优质地处理，再强的算法也不能为我所用。

在接下来的学习中，在训练营给予的目标之外，我会偏向于重点学习如何进行数据预处理和特征工程，以更好地发挥一般算法，这样再往更高级的算法学习和应用，这会是一段需要积累和学习的过程。

这次的数据挖掘的过程让我比较清楚地体会处理的细节。在 pandas 的使用的时候，它非常方便，功能很强大，有挺多方法也是在数据处理的时候学习到的，同时也会有出现问题的时候；另外，也还使用到了正则表达式中地正向先行断言和正向后行断言，体会到了用来提取时十分便捷；而挖掘过程感到是曲曲折折地，还有有很多时候是需要退 1 步，重新思考，特征与目标值的关系并不是想当然的，还是需要从全面地使用数据去说明事实，并不是很容易能够完成，需要更加沉稳的心和清晰的思维。这很重要。快乐！

补充*：过程有些反复，数据删了后又重新分析，代码文件会有些杂乱。大体的挖掘过程如上。辛苦翻阅！

*詹培林（2001—），男，广东工业大学 2020 级计算机类专业在读。