数据分析及实践实验五实验报告

学号：PB21000224　　姓名：陈鸿绪　　日期：6.1

一 .实验内容

根据实验三的特征工程，使用 PISA2018 数据集，使用至少一种分类方法（如：决策树、KNN、朴素贝叶斯或者感知机、集成算法等），测试算法对 PISA2018 数据集“REPEAT”特征的预测功能。具体预测好坏评价指标有：accuracy 和 F1-score。

二 .实验处理、分类方法、预测评价指标介绍：

1. **KNN 模型预测：**KNN 算法本质是监督学习算法，可以用于分类和回归问题，具体是对于新的输入实例，通过已经训练的数据集找到与该实例最近的 K 个邻居，再通过距离度量来确定属于哪一个类别，特别地二元预测时 K 值为 2。而在 python 的 sklearn 模块中已经集成了 KNN 算法处理程序。所以在本实验中采用了将测试数据输入到 sklearn 的 KNN 分类器中进行训练，最后使用预测数据集再进行对 REPEAT 列的预测。
2. **多层感知机预测：**这是监督学习算法，通过训练数据集梯度下降求得输入层、输出层、隐藏层，隐藏层可以由一个或者多个非线性层组成。本次实验实际上可以认为是一个二分类任务，而 sklearn 模块中有神经网络的部分，所以只需要进行训练和调参即可。
3. **实验数据取样方法：**由于该数据集的正负分布不均匀，所以为了让预测结果不具有明显偏向性，对训练数据采用了取样方法：过采样、SMOTE 采样、ADASYN 采样。如其中的过采样，实际上就是通过重复选择某些样本使得正负比例近似 1：1 避免预测偏向性。对于这些采样，可以引入 imblearn.under_sampling，其中包含了这些采样方法。
4. **模型验证方法：**本实验采用交叉验证法，将数据集分层分为 k 个大小几乎一样的互斥子集的并集作为训练集，剩下的子集为测试集，每次验证返回评价指标，最后计算指标的平均值。在本实验中采用 k=5，即划分为 5 折。
5. **模型评价指标：**实际上，单单用 accuracy 作为评价指标是远远不够的，比如，在本数据集中，正负样本数量比例大致 1：4，如果训练模型直接取所有情况预测都是 0，那么准确率仍然可以达到 0.8，但是一旦更换数据集这种预测模型就会毫无意义，所以有必要采用另一个评价指标 F1-score。两者都可以调用 sklearn 模块中的函数来进行计算。

三 .实验过程

由于实验过程繁琐复杂，牵扯到非常多的试错过程，所以笔者就分比较重要的几个步骤来介绍实验过程。这些过程目标很明确，就是在正确的同时保证 acc 与 F1-score 尽可能的大。

1.过采样的 KNN 模型预测：

模型解释：该预测模型对应于源代码文件中的 KNN_1()函数，具体来说，该函数先选出自变量特征列、因变量特征列所对应的数据分别为 X、y，然后通过数据预处理将缺失值补全，再使用 StratifiedKFold 函数，即使用分层 kfold，将数据集分为 5 折。4：1 轮换划分给训练集与测试集。由于原数据正负分布不均匀，采取过采样方法，将 X_train 中某些样本被多次选择，从而达到训练集正负比例相同的效果。再将过采样后的数据对 KNN 分类器进行训练，训练完毕预测剩下的测试集，打印五次所得 Accuracy 和 F1-score。函数返回 F1-score 的平均值和 accuracy 的平均值。

模型预测结果：选取特定特征并尝试组合（其实很多特征都是通过实验三特征工程中得出的弱相关特征），得到如下结果（接下来约定[word_lis]:[a,b]意思为：该模型选取 word_list 作为自变量特征集合，a 为预测值和测试集本身的值之间的 acc，b 为两者之间的 F1-score）从上而下按顺序进行：

['OCOD1','OCOD2','OCOD3'] : [0.7277359963776212, 0.2910235469527633]

分析：F1 太小，不理想

['COBN_S' ,'COBN_M' ,'COBN_F']：[0.7738285632214438, 0.1192408133006739]

分析：F1 太小，依然不理想

["PROGN"]：[0.8047705615951957, 0.14096430600613263]

分析：F1 太小，依然不理想

['ST023Q01TA','ST023Q02TA','ST023Q03TA','ST023Q04TA','ST023Q05TA']：

[0.7896671026000572, 0.034546071826525485]

分析：F1 几乎为 0，该特征直观上不能与其他特征组合预测。

['SCCHANGE']：[0.7900701034856963, 0.05151634501772352]

分析：F1 几乎为 0，该特征直观上不能与其他特征组合预测。

['MISCED','FISCED','HISCED','PARED','MISCED_D','FISCED_D','HISCED_D','PAREDINT','BMMJ1','BFMJ2','HISEI']：[0.7259817453243966, 0.25320543716561617]

分析：F1 太小，依然不理想

['ISCEDL']：[0.49670430421686157, 0.4206823529107238]

分析：虽然 acc 只有 0.5 左右，但是这个特征在 F1 上的优势非常明显，后面组合特征大概率会将其加入其中

['OCOD1','OCOD2','OCOD3','PROGN']：

[0.7382633268154125, 0.32363892640822306]

分析：这是上面的 F1 并不理想的两个特征集之并(['OCOD1','OCOD2','OCOD3']['PROGN'])，发现两者结合后 F1 值有所提升，初步判断猜想：对应的两个 F1 值不是非常小且相差不是很大的两个特征集组合在一起会使得新得到的 F1 提升。按此思路继续。

['COBN_S' ,'COBN_M' ,'COBN_F','PROGN']：

[0.7917772675492701, 0.23924439960452007]

分析：对比前后 F1 大小，初步判断猜想大部分情况都正确，继续组合。

['COBN_S' ,'COBN_M' ,'COBN_F','PROGN','OCOD1','OCOD2','OCOD3']：

[0.7407767090710966, 0.34260413905649445]

分析：同样发现 F1 值有所提升。

['PROGN','COBN_S' ,'COBN_M' ,'COBN_F','OCOD1','OCOD2','OCOD3','ISCEDL']:

[0.7663362819969065, 0.4124269984193346]

分析：发现 F1 已经提升至至少 0.41 左右的水平

['ST005Q01TA','ST007Q01TA','ST011D18TA','ST011D19TA','ST013Q01TA''IC150Q06HA','IC150Q09HA','IC151Q06HA''IC151Q08HA','IC151Q09HA','IC152Q06HA','IC152Q08HA','IC152Q09HA','EC150Q06WA','EC151Q03WA','EC152Q01HA','EC153Q02HA','EC159Q01HA','FL164Q01HA', 'OCOD1', 'OCOD2','OCOD3', 'PROGN','ISCEDL','BMMJ1','BFMJ2','HISEI','BSMJ''CHANGE''ESCS','HOMEPOS','INFOCAR']：

[0.7983213613263842, 0.4752237124238075]

分析：这是后面通过神经网络二分类方法得到的 F1 值较优的单个特征值所有的集合，发现 F1 水平在 0.48 左右，但仍然不能称得上为优。

此时做到这里，我只能从其他方向入手，比如改变采样方式，而非单一过采样方法。

2. SMOTE 采样的 KNN 模型预测：

模型只是将 KNN 算法中过采样改成为 SMOTE 采样。得到预测结果：

['PROGN','COBN_S' ,'COBN_M' ,'COBN_F','OCOD1','OCOD2','OCOD3','ISCEDL']：

[0.7766738703886553, 0.4065785878341391]

分析：其实横向对比基于过采样的 KNN，两者几乎没有显著变化。甚至多次测试后 SMOTE 大致比过采样的 F1 少了 0.01 左右，即没有任何改进。

['ST005Q01TA','ST007Q01TA','ST011D18TA','ST011D19TA','ST013Q01TA''IC150Q06HA','IC150Q09HA','IC151Q06HA''IC151Q08HA','IC151Q09HA','IC152Q06HA','IC152Q08HA','IC152Q09HA','EC150Q06WA','EC151Q03WA','EC152Q01HA','EC153Q02HA','EC159Q01HA','FL164Q01HA', 'OCOD1', 'OCOD2','OCOD3', 'PROGN','ISCEDL','BMMJ1','BFMJ2','HISEI','BSMJ''CHANGE''ESCS','HOMEPOS','INFOCAR']：

[0.7924886147183592, 0.5102747270699347]

分析：横向比较，确实有非常大的改进，F1 已经达到了 0.51 的水平

3.ADASYN 采样的 KNN 模型预测：

模型只是将 KNN 算法中过采样改成为 ADASYN 采样。得到预测结果：

['PROGN','COBN_S' ,'COBN_M' ,'COBN_F','OCOD1','OCOD2','OCOD3','ISCEDL']：

[0.768233154378601, 0.4060438847767175]

分析：多次实验横向比较发现，几乎没有改进。

['ST005Q01TA','ST007Q01TA','ST011D18TA','ST011D19TA','ST013Q01TA''IC150Q06HA','IC150Q09HA','IC151Q06HA''IC151Q08HA','IC151Q09HA','IC152Q06HA','IC152Q08HA','IC152Q09HA','EC150Q06WA','EC151Q03WA','EC152Q01HA','EC153Q02HA','EC159Q01HA','FL164Q01HA', 'OCOD1', 'OCOD2','OCOD3', 'PROGN','ISCEDL','BMMJ1','BFMJ2','HISEI','BSMJ''CHANGE''ESCS','HOMEPOS','INFOCAR']：

[0.7829570871217518, 0.5027396581936558]

分析：横向比较，F1 值比过采样优，但是不及 SMOTE

4.基于 KNN 模型的实验结果总结：

总的来说，在尝试了很多种特征组合和采样方法后发现：基于 SMOTE 采样的 KNN，特征采取['ST005Q01TA','ST007Q01TA','ST011D18TA','ST011D19TA','ST013Q01TA''IC150Q06HA','IC150Q09HA','IC151Q06HA''IC151Q08HA','IC151Q09HA','IC152Q06HA','IC152Q08HA','IC152Q09HA','EC150Q06WA','EC151Q03WA','EC152Q01HA','EC153Q02HA','EC159Q01HA','FL164Q01HA', 'OCOD1', 'OCOD2','OCOD3', 'PROGN','ISCEDL','BMMJ1','BFMJ2','HISEI','BSMJ''CHANGE''ESCS','HOMEPOS','INFOCAR']这个集合，F1 值达到一个相对而言比较优的值：0.51 左右。Acc 也处于一个不低的水平。

然而在实际应用中，F1 为 0.51 水平并不是非常好，一般而言，F1 需要达到 0.6 以上的水平才可以称得上预测是可能有效的。所以 KNN 模型在我尝试的范围并没有达到一个好的 F1 值。

5.基于神经网络的多层感知机模型：

类似的，我们可以通过相关性来直接挑选特征，但这往往不是非常准确，有的时候相关性为弱相关的特征 F1 跑出来也非常小。所以这里采用了一个比较暴力的手段，通过所有特征一一输入神经网络得到单个特征预测所对应的 F1 值，如果 F1 值大于一定阈值（实验中取的是 0.37），则将该单个特征纳入列表中，依靠之前的猜测我们可以将这些大于阈值的单个特征

组合在一起、排开实验硬性排除的强相关特征，如此 F1 应该有大幅提升。实际上也确实如此，经过暴力挑选出来的特征在经过调参后确实可以让预测后 F1 达到 0.61 水平。

该暴力挑选出来的特征集合为：

S=['ST005Q01TA','ST007Q01TA','ST011D18TA','ST011D19TA','ST013Q01TA''IC150Q06HA','IC150Q09HA','IC151Q06HA''IC151Q08HA','IC151Q09HA','IC152Q06HA','IC152Q08HA','IC152Q09HA','EC150Q06WA','EC151Q03WA','EC152Q01HA','EC153Q02HA','EC159Q01HA','FL164Q01HA', 'OCOD1', 'OCOD2','OCOD3', 'PROGN','ISCEDL','BMMJ1','BFMJ2','HISEI','BSMJ''CHANGE''ESCS','HOMEPOS','INFOCAR']

接下来任务就是通过调参或者调整某些特征，让 F1 尽可能变大到 0.6 的水准，acc 可以保持在 0.8 左右的水准。以下为具体操作：

选择自变量特征集合 S，

设置神经网络参数为：hidden_layer_sizes=(40, 30)

[acc,F1]= [0.7927259552942015, 0.5891811346899444]
F1 为 0.59 左右，尚未达到水准 0.6。仍需调参。

设置神经网络参数为：hidden_layer_sizes=(20, 30)

[acc,F1]= [0.7974678284811502, 0.6119160129567353]
F1 达到 0.61，且 acc 也几乎为 0.8，可以说明此时是一个比较优的预测

设置神经网络参数为：hidden_layer_sizes=(10, 30)

[acc,F1]= [0.7983923838979529, 0.619971074340665]
F1 达到 0.61，且 acc 也几乎为 0.8，可以说明此时是一个比较优的预测，继续调参

设置神经网络参数为：hidden_layer_sizes=(10, 40)

[acc,F1]= [0.7972308448591479, 0.6192997610250399]
这个预测稍稍差于 hidden_layer_sizes=(10, 30)

设置神经网络参数为：hidden_layer_sizes=(4, 30)

[acc,F1]= [0.7926072217664268, 0.6213893055134319]
这个预测 F1 已经达到了 0.621 的水平了

设置神经网络参数为：hidden_layer_sizes=(3, 29)

[acc,F1]= [0.7939824300012113, 0.622342343372086]

6.基于神经网络的多层感知机模型预测结果总结：

其实笔者不止调参了这几个，只是综合情况而言 hidden_layer_sizes=(3, 29)确实是相对比较优的参数了，一方面其的正确率达到了 0.79 左右，另一方面 F1-score 也能稳定在 0.62 左右，几乎全部在 0.61 以上，所以，对于该神经网络而言，（3，29）与 ['ST005Q01TA','ST007Q01TA','ST011D18TA','ST011D19TA','ST013Q01TA''IC150Q06HA','IC150Q09HA','IC151Q06HA''IC151Q08HA','IC151Q09HA','IC152Q06HA','IC152Q08HA','IC152Q09HA','EC150Q06WA','EC151Q03WA','EC152Q01HA','EC153Q02HA','EC159Q01HA','FL164Q01HA', 'OCOD1', 'OCOD2','OCOD3', 'PROGN','ISCEDL','BMMJ1','BFMJ2','HISEI','BSMJ''CHANGE''ESCS','HOMEPOS','INFOCAR']特征集合对于该数据集而言可以较好的预测 REPEAT，正确率在 0.8 左右，F1-score 在 0.62 左右。

四．实验结论

1. 在 KNN 模型下，笔者采用了多种不一样的采样方式，以此来抵消样本不平衡的影响，也没有将预测的 F1 达到 0.6 以上，最多也只有 0.51，可以认为这是一个比较失败尝试。
2. 在基于神经网络的多层感知机模型预测下，笔者通过暴力方法找出所有单个预测所得到的 F1 值较大的特征，将其作为整体输入训练神经网络，后续经过一系列调参得到了一个相对可以使 F1 值稳定在 0.62 左右、acc 稳定在 0.8 左右的预测模型。
3. 对于两种不同算法模型，可以发现，KNN 作为比较传统的方法预测时间较短，但预测可靠度不高，然而对于基于神经网络的多层感知机模型而言，其训练的计算代价比较大，预测时间也相对而言更长，然而其预测效果还是显著优于 KNN 模型的。

[参　考　文　献]

1. CSDN,多层感知机（MLP）实现考勤预测二分类任务（sklearn）
2. 知乎,python sklearn 中 KFold 与 StratifiedKFold
3. CSDN,特征锦囊：如何在 python 中处理不平衡数据