机器学习课程大作业

题目：基于机器学习的经济困难生精准资助研究

吕之豪　　1810064

崔嘉珩　　1710018

戴文娇　　1810151

肖韵竹　　1810189

2020 年 12 月 27 日

摘　要

国家近年建立健全资助政策体系，其中经济困难学生的认定尤为重要，各地、各高校对其也尤为重视。但通常由于资助资金有限，并且经济困难学生的个人信息和情况各不相同，要进行精准资助有一定难度。因此，我们希望利用机器学习模型来对这个问题进行解决。

本课题通过先基于有监督学习和本校现有经济困难生相关数据，建立适合本校实际情况的有监督三分类模型。再对少量数据做强标签标注，产生少量多等级分类数据，借助半监督学习方法训练出细粒度的多分类模型，实现适合全部数据的、对经济困难学生情况的多等级评估模型，使经济困难生的认定更加准确，实现精准资助。该模型将用于帮助本校各个学院进行经济困难生的评审等问题，辅助专家系统进行决策。根据数据的叠加以及模型的逐步完善，可将模型进一步推广和精确，普适地用于各学校经济困难生的认定工作。

关键词：经济困难生认定；特征工程；有监督学习；半监督学习

一、项目背景与简介

（一）项目背景

2007 年 5 月，国务院颁发了《关于建立健全普通本科高校、高等职业学校和中等职业学校家庭经济困难学生资助政策体系的意见》，决定从 2007 年秋季学期开学起，建立健全我国高校家庭经济困难学生资助政策体系。新资助政策体系建立后，国家在高等教育阶段基本建立起国家助学金、国家励志奖学金、国家奖学金、国家助学贷款、师范生免费教育、勤工助学、特殊困难补助、学费减免和“绿色通道”等多种形式有机结合的家庭经济困难学生资助政策体系。

2007 年，国务院规定国家助学金资助面平均约占全国普通本科高校和高等职业学校在校生总数的 20%。平均资助标准为每生每年 2000 元，具体标准由各地根据实际情况在每生每年 1000-3000 元范围内确定，可以分为 2-3 档。

政府、高校、社会各界的资助，为经济困难学生安心学习顺利完成学业提供了保障。2019 年中国学生资助发展报告中显示，各类助学金共资助全国普通高校学生 1130.66 万人次，资助金额 378.29 亿元，占高校资助资金总额的 28.73%。国家助学金共资助学生 928.78 万人，资助金额 345.28 亿元。[1]

从以上数据可以看出，经济困难学生的认定尤为重要，各地、各高校对其也尤为重视。但通常由于资助资金有限，并且经济困难学生的个人信息和情况各不相同，要进行精准资助有一定难度。

（二）国内外研究现状

由于资助资金有限，并且经济困难学生的个人信息和情况各不相同，要进行精准资助有一定难度。现有的高校经济困难学生的认定多是专家系统模式——以预先人为指定的标准和权重进行量化，然后面向困难生开展评审，进行综合判定。而人为预先设定权重存在对实际参评的学生缺乏了解，标准一刀切；纯靠专家的经验确定指标，缺乏相关依据，一定程度影响评审的准确性和公平性等问题。

针对专家系统的弊端，国内外许多专家学者提出了一些计算机算法，用于脱离系统专家评审来进行判断，辅助专家决策：

1.陆孙琦[2]基于 S 校的调研数据，分析原因，并提出解决方案。张善红[3]构

建了高校助学金评定量化模式，并通过实例研究对该模式进行检验，为以后进一步研究提供依据。刘美荣[4]针对高校助学金申请、评审及监督过程中存在的问题，提出了相应的解决建议。

2. 美国联邦政府早在 1992 年就建立了联邦计算法（FM 计算法），用于认定家庭经济困难学生，并将其写入高等教育法，一直沿用至今。据美国大学委员会和学生资助管理者协会的调查，全美超过 70%的高校直接使用 FM 计算法，其余高校修改使用。[5]

3. 在美国建立相关算法后，我国也开始着力研究判定经济困难学生的相关问题。欧阳铁磊、叶玲肖在《基于大数据分析的高校贫困生精准资助策略研究》[6]一文中指出采用 CHAID 算法，找出最佳分组变量和分组点，设计判别经济困难学生的模型，为学校资助经济困难学生提供数据参考。

近年来，依托大数据技术发展和计算能力的提升，以机器学习构建的分类决策模型精度逐步提高，在一定程度上解决了经济困难学生的认定问题，但收集到的经济困难生的信息一般是非结构化数据，并且数据量有限，难以进行数据处理，无法直接套用机器学习，在精准资助方面的效果并不好。

（三）项目简介

本项目先基于本校现有经济困难生相关数据与通过数据增强产生的非困难生数据，建立适合本校实际情况的有监督三分类模型。再令专家对少量数据做强标签标注，产生少量多等级分类数据，借助半监督学习方法训练出细粒度的多分类模型，实现适合全部数据的、对经济困难学生情况的多等级评估模型。实际运行时先用有监督模型判断该生属于“非困难生”、“一般困难”还是“特别困难”。若其不为“非困难生”，则利用半监督模型对其困难程度进行更细致的判断，使经济困难生的认定更加准确，实现精准资助。

半监督学习的优势在于：只需要令专家进行少量数据的评定与标注，因此其比有监督学习开销小，同时也比无监督学习更加准确，因此使用半监督学习可以比普通的算法或有监督学习方法得到更好的结果。同时，半监督学习的前景广阔，而现有的应用却并不多，本项目实现的模型有可观的应用前景，因此可以验证半监督学习相关理论的实用性。

二、数据预处理与特征工程

（一）数据概况与预处理

项目的原始数据为本校 2017-2019 年经济困难生的申请表，其中包括共 8333 条数据。经人工考察，共有 7 条数据填写出现问题，且难以进行人工处理，故删去。

申请表具有如下 15 个特征：“院系”、“专业”、“民族”、“出生年月”、“享受国家政策资助情况”、“入学前户口性质”、“家庭主要经济来源”、“家庭人口”、“家庭年收入”、“家庭其他成员在受教育情况”、“突发事件情况”、“是否贷款”、“在校受奖励资助情况”、“所在校区”、“备注”，标签为“院系认定贫困类型”。其中“院系”、“专业”、“出生年月”、“所在校区”、“备注”对模型判断困难类型无帮助，故删去。

经过以上预处理后，数据集共有 8326 条数据，10 个特征：“民族”、“享受国家政策资助情况”、“入学前户口性质”、“家庭主要经济来源”、“家庭人口”、“家庭年收入”、“家庭其他成员在受教育情况”、“突发事件情况”、“是否贷款”、“在校受奖励资助情况”，标签：“院系认定贫困类型”。

为了半监督学习模型能够成功运行，我们需要得到少量的样本，令专家系统进行人工细化标签处理。原数据按照年份-学院-专业的顺序进行排序，为了使样本集合能够较好地代表整个数据集，我们将数据随机打乱。接着从中随机抽取 400 个数据，发给辅导员和专家评审组，进行标签的细化，由以前的“一般困难”、“特别困难”细化为共四个等级，按照困难程度由高到低排序为 1 至 4。

（二）目标形式

本校于 2020 年开始使用新的网上学生资助管理系统，该系统将大部分特征均进行了结构化处理。我们意图将 2017-2019 年的非结构化数据整理为现有系统中的形式，最终的特征与变量类型请见附录 A 表 A.1。

除了将特征结构化外，还需要对标签进行整理。其中有监督模型和半监督模型所用到的数据标签不同，如附录 A 表 A.2 所示。

（三）特征工程

特征和标签在数据集中的情况以及具体处理方法见下，其中关键词检索为一常用的方法，对于每一列都需要人工查看数据，根据目标结果总结出该列的关键词词典，具体的词典请见代码。

注：以下出现的数据均只代表原数据中有类似的格式，而并非真实数据，真实数据为保证学生个人隐私无法公开。

1. 民族：

数据情况：有空值，其余值均为一字符串，含有该学生的民族。

处理方法：先用“汉”填充空值。再对所有数据识别关键字“汉”，若有则将其值设为 0；否则设为 1。

2. 享受国家政策资助情况：

数据情况：有空值，其余值均为一字符串，多为几个词或一句话，描述其受的国家政策。

处理方法：先用“无”填充空值。对于所有的选项，每个新建一列，再对所有数据识别该选项的关键字，如在填充“城乡低保户”一列时搜索“低保”，若有则将相应的列值设为 1；否则设为 0。之后抛弃“享受国家政策资助情况”一列。

3. 入学前户口性质：

数据情况：有空值，其余值均为字符串，除“农业”、“城镇”外有大量格式不规范的数据，如：“城市”、“非农村”、“非农业”等。

处理方法：先用“城镇”填充空值。再对所有数据识别关键字“城”、“非农”、“居民”，若有，则将其值设为 0；否则设为 1。

4. 家庭主要经济来源：

数据情况：有空值，其余值均为字符串，该列符合目标的数据较少，大部分数据为一个词或短语，描述经济主要来源；少部分数据为多个词或一句话，描述了多个经济来源。

处理方法：先用“父母均下岗”填充空值。对于所有的选项，每个新建一列，再对所有数据识别该选项的关键字，如在填充“做生意”一列时搜索“生意”、“从商”等，若有则将相应的列值设为 1；否则设为 0。之后抛弃“家庭主要经

济来源”一列。

5.家庭人口：

数据情况：有空值，一部分数据为整数，不规范的数据为字符串，如“3 人”、“3 口人”、“四”等。

处理方法：先用 3 填充空值。再将 0-9 的阿拉伯数字和中文数字设为关键词，进行识别，筛选出每个 1-9 人口数所对应的数据，并用相应数字进行替换；由于辅导员和专家组认为人口数大于等于 10 的数据均为异常值，最后筛选出人口大于等于 10 的数据，并将其剔除。

6.家庭年收入：

数据情况：有空值，一部分数据为浮点数，不规范的数据为字符串，如“3 万元”、“不到 3 万元”、“7000-10000”、“收入不固定”、“暂无”。

处理方法：先用 0 填补空值。由于本特征较为复杂，选择手动进行处理：筛选出字符串型的数据，将形如“3 万元”的改为“30000”，形如“不到 3 万元”的改为“30000”，将形如“7000-10000”的取区间中值，将将形如“收入不固定”的改为此行数据的家庭人数*11760（11760 为每人每年的低保金额），形如“无”的改为 0。此时将上述数据除以对应家庭人数，并输出至“家庭人均年收入”一列，抛弃“家庭年总收入”这一列。

7.家庭其他成员在受教育情况：

数据情况：有空值，非空数据全部与目标格式不同，处理起来非常困难。有的数据为一字符串，有的数据为一个整数。字符串中，大部分数据在介绍自己家有几个人在上什么学校，如“1 人高中 1 人大学”、“妹妹高中”、“两个妹妹分别在上高一、高二”；少部分数据还写了自己家人的受教育情况，如“爸爸大学毕业，妈妈初中文凭，哥哥和我在读大学”。

处理方法：先用“无”填充空值。再对所有数据用 Python 的 jieba 库[7]进行分词操作（若发现分词结果与预想中有差别，则用 jieba.suggest_freq()函数对分词字典进行修改）。遍历分词结果：若某个词是关键词，则在 cut_type 这一 list 对象中记录其“词性”（见附录 A 中表 A.3）；若不是关键词则不记录。之后对 cut_type 进行遍历，进行关键词模式的识别，具体的模式组和对应处理方法见附录 A 中表 A.4。最后将模式组识别的结果填充到“大学在读人数”、“高中在读人数”、“义

务教育在读人数”三列中，并抛弃“家庭其他成员在受教育情况”一列。

8.突发事件情况：

数据情况：有空值，非空数据全部与目标格式不同，处理起来非常困难。数据全为字符串，少部分数据填写的内容与系统给定的关键词一样，大部分数据写的非常细致，如“奶奶冠心病，常年卧床。爷爷去世 妈妈下岗，没有工作”、“去年庄稼受到百年难遇洪水导致颗粒无收”。

处理方法：先用“无”填充空值。检测是否有自然灾害情况，利用关键词识别，若数据中出现关于灾害的关键词则将“突发重大自然灾害”一列设为1。再识别孤残情况，若利用关键词识别，若数据中出现关于灾害的关键词则将其“享受国家政策资助情况”中“孤残学生”一列设为1。再对所有数据用jieba进行分词操作（这里也可能对分词字典进行修改）。遍历分词结果：若某个词是关键词，则在cut_type这一list对象中记录其“词性”（请见附录A中表A.5）；若不是关键词则不记录。之后对cut_type进行遍历，进行关键词模式的识别，具体的模式和对应处理方法见附录A表A.6。最后对于每个选项都创建一新列，将模式识别的结果填充到其中，并抛弃“突发事件情况”一列。

9.是否贷款：

数据情况：有空值，数据均为字符串，一部分数据为“是”或“否”，不规范的数据形如“是，有助学贷”、“生源地贷款”、“有房贷”、“30000”等。

处理方法：先用0填充空白值，然后通过关键词筛选出助学贷的标签并将其值设为1；否则将其改为0。

10.在校受资助奖励情况：

数据情况：有空值，其余值均为一字符串，大多为几个长的词，每个词为一个助学金。我们从辅导员处获取到了能够合法填在此列的助学金类型和相应的金额，并发现原数据中有很多资助获奖励对于这列的标准是不合法的，需要剔除。

处理方法：先用“无”填充空值。增加“获得助学金个数”、“获得助学金金额”、“获得国家助学金情况”三列，将其全部填充为0。再对所有数据识别每个助学金的关键字，若有，则该数据的助学金个数加1，助学金金额加对应的金额；同时若该助学金为国家一等助学金，则将获得国家助学金情况赋值为2，若该人未获得过国家一等助学金，但监测到获得过国家二等助学金，则将获得国家助学

金情况赋值为 1。最后抛弃“在校受资助奖励情况”一列。

11.院系认定贫困类型：

数据情况：细化标签均为取值在{0,1,2,3}中的整数，因此不需要处理；原标签无空值，均为字符串，大多数数据为“一般困难”和“特别困难”，不规范的数据形如“一般贫困”、“特殊困难”等。

处理方法：通过关键词“特”和“一般”分别筛选出“特别困难”和“一般困难”所对应的数据，并用 2 和 1 进行替换。

三、数据增强

（一）为什么要进行数据增强

现有的数据只含有“一般困难”与“特别困难”的数据，并无“非困难”的数据。若仅用现有数据集对模型进行训练，得到的系统在未来进行预测时，无法识别非困难学生，反而会将其判定为一般困难学生甚至特别困难学生。这样的系统鲁棒性很差，不能用于未来使用。因此我们需要将非困难生的数据加入数据集。

（二）数据增强的方法

由于我们无法得到本校非困难生的数据，我们只能通过一些准则构建出非经济困难生的数据。我们通过咨询辅导员和评审专家，了解到一般情况下非经济困难生和经济困难生的相同点和不同点，并构建了如下非困难生数据的生成方式，生成共 1000 个非困难生数据。

1. 家庭人均年收入：

人均收入总体分布应该是正态的，但是由于少数高收入人群的影响，实际应当是厚尾（右偏）的。由于后续用到的学习模型涉及到联合分布概率或者参数距离，比较适合进行正态分布转化，因此我们决定用对数正态分布。该对数正态分布要满足转化为正态分布后均值为 26400（天津市人均可支配年收入），且 0.01 分位数为 11760（天津市每人每年低保金），由此得到均值为 10.1811，标准差为 0.1892。即我们使用对数正态分布 Lognormal(10.1811, 0.0358) 作为家庭总收入的模拟分布，其图像见图 3.2.1。

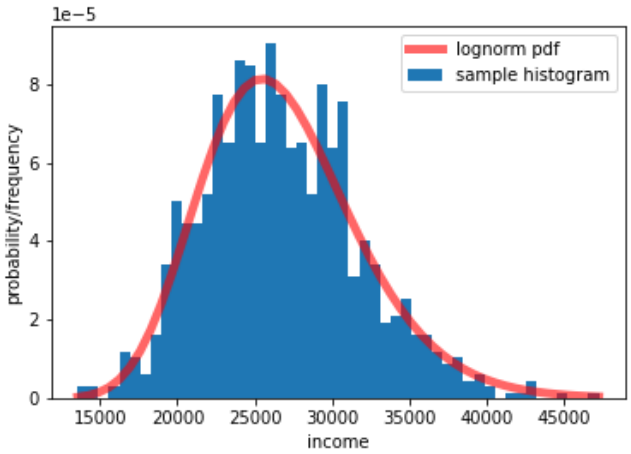

图 3.2.1　家庭人均年收入的模拟抽样分布 Lognormal(10.1811, 0.0358)

2. 建档立卡贫困户、城乡低保户、五保户、孤残学生、低保、助学金个数、助学金金额、国助类型、是否贷款：

这些特征在非困难生中均设为 0，因为这些特征表示家庭硬性条件较为困难，在实际使用中，学校会对有上述特征的人进行实现调查，并确保对他们进行困难生补助。

3. 军烈属或优抚子女、父母均下岗、父母一方下岗、大学、高中、义务教育、祖父母患病、父母离异、父亲（母亲）患普通疾病、父母患普通疾病、兄弟姐妹患重疾、父亲（母亲）患重疾、父母患重疾、父亲（母亲）去世、突发重大自然灾害、民族、家庭人口、入学前户口性质：

利用二项分布来构造数据。先统计非困难生每个特征中不同值的比例，并根据辅导员提供的情况进行微调。二项分布的参数(n, p)见附录 A 的表 A.7。

四、数据降维与可视化探究

（一）用主成分分析进行降维

现在我们得到了经过所有特征处理的数据集，发现一共有 32 个特征。一个很自然的想法为利用主成分分析对其进行降维，变为低维数据，再喂给机器学习模型。

利用 Python 中 sklearn 库[8][9] decomposition 模组中的 PCA 方法，对数据进行主成分分析。分析结果发现将特征值从大到小排序后，前 10 个主成分的方差贡献率共计约 81.7%（前 10 个主成分的方差贡献率见表 A.8），这说明无法找到较少的主成分使其充分代表数据的特征。因此不能用主成分分析对其进行降维，我们需要考虑其他方法。

（二）用 t-SNE 算法进行高维数据可视化

为了找出 PCA 失效的原因，我们需要对数据集进行可视化。但该数据集的维度为 32，是无法直接作图的。为了作出三维图像，我们需要利用 t-SNE (t-Distributed Stochastic Neighbor Embedding)算法。

1. SNE 算法[10]**：**

SNE 算法的基本思想为高维空间距离相近的数据点，映射到低维空间距离也是相近的。其用条件概率来描述两个数据点之间的距离（注意：该“距离”并不满足对称性），定义x_j相对x_i的条件概率$p_{j|i}$为以x_i为正态密度均值，x_j作为x_i的临近值被抽中的概率，即：

$$p_{j|i} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \tag{4.1}$$

其中σ_i^2需要根据选择的x_i进行二分搜索[10]来确定。在确定σ_i^2后，当x_j与x_i的欧氏距离较小时，$p_{j|i}$会相对较大；反之当数据非常稀疏时，$p_{j|i}$会变得极小。令x_i，x_j在低维空间对应的点分别为y_i，y_j，相似地，我们可以定义y_j相对y_i的条件概率$q_{j|i}$为：

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)} \tag{4.2}$$

为了保证高维空间距离相近的数据点，映射到低维空间距离也是相近的，我们需要用 KL 散度来衡量两个分布之间的距离：

$$C = \sum_i KL(P_i|Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{4.3}$$

我们希望两个分布之间距离较小，因此需要找$\mathcal{Y} = \{y_i\}_i$使得C最小。我们可以使用梯度下降法找到对应的$\mathcal{Y}$，若$\mathcal{Y}$中所有元素均为三维向量，则可以画出其图像。但 SNE 算法只考虑将类内距离缩小，不考虑类间距离，因此所有的数据在投影后，很可能间距会非常小，可视化效果差。

2. t-SNE 算法理论[11]**：**

t-SNE 算法是对 SNE 算法的优化。该算法让$p_{j|i}$和$q_{j|i}$具有对称性（请注意分母求和的角标），并记为p_{ij}与q_{ij}，具体定义为：

$$q_{ij} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq l} \exp(-\|y_l - y_k\|^2)} \tag{4.4}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{4.5}$$

根据此定义，所有数据在投影后，类内间距尽可能小，类间间距尽可能大。这样找到使得C最小的三维向量集合$\mathcal{Y}$可视化效果较好。

3. t-SNE 算法的使用：

我们使用 Python 中 sklearn 库 manifold 模组中的 TSNE 方法，对所有数据进行 3 维可视化。迭代至稳定，迭代次数为 500 次，得到图 4. 2. 1。

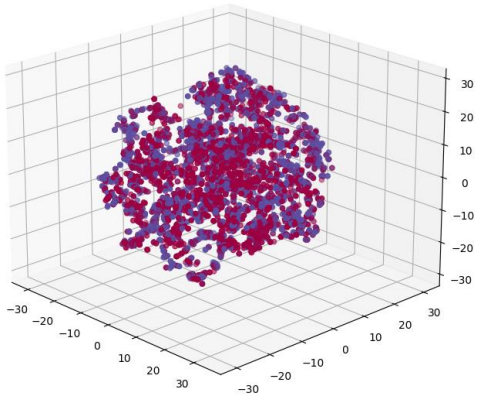

图 4. 2. 1　t-SNE 算法对所有数据可视化结果

可以看到数据在三维空间中为一个球体，且“一般困难”和“特别困难”的数据互相包裹在一起（像 Swiss Roll 数据那样），这说明本数据集不适合降维，只能在高维空间下才能比较容易分离。因此我们不改变数据集的特征，直接喂给机器学习模型。

（三）用 t-SNE 算法进行标签细化数据的可视化

我们也可以用 t-SNE 算法对标签细化后的数据进行可视化，来检验标签的合理情况。标签细化时，专家组决定困难程度 1 和 2 必须为“特别困难”的学生，困难程度 3 和 4 必须为“一般困难”的学生。我们分别对这两组进行可视化。

首先对“特别困难”的数据进行可视化，迭代收敛后得到图 4.3.1，局部放大后得到图 4.3.2：

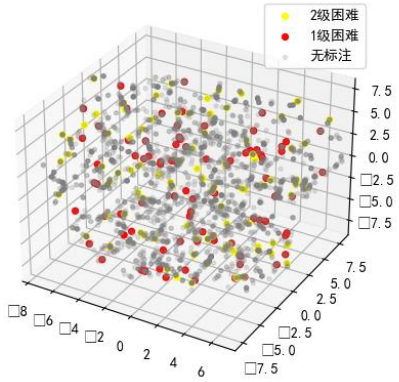

图 4.3.1　t-SNE 算法对特别困难数据可视化结果

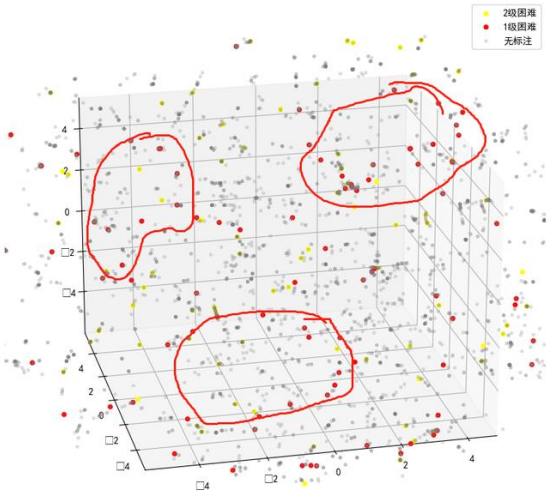

图 4.3.2　t-SNE 算法对特别困难数据可视化结果（放大后）

可以看到“非常困难”的数据和总数据集的情况类似，都是线性不可分的，

需要分类、拟合能力强的模型来对其处理。但是在放大之后可以看出局部是有一些小的聚类的，因此可以认为其满足半监督模型的聚类假设。

我们可以对“一般困难”的数据也做类似的处理，并得到类似的结论（结果图为图 4.3.3，放大后为图 4.3.4），并且可以看到“一般困难”的聚类更明显，也满足聚类假设。

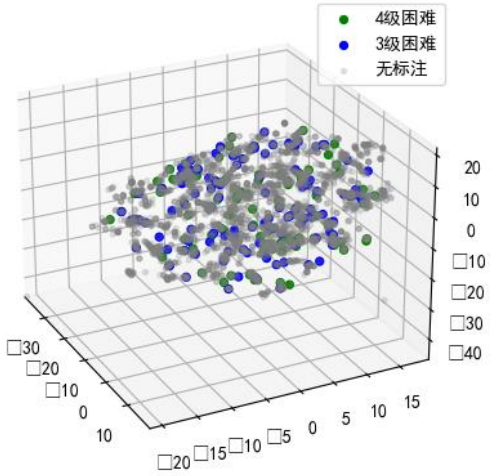

图 4.3.3　t-SNE 算法对一般困难数据可视化结果

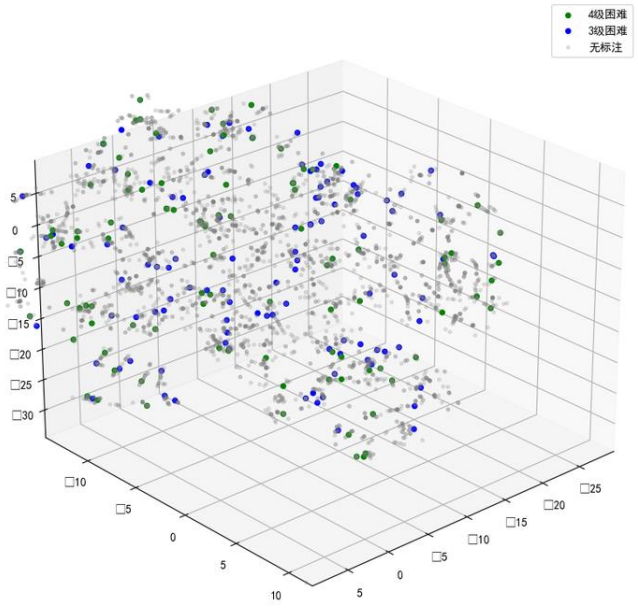

图 4.3.4　t-SNE 算法对一般困难数据可视化结果（放大后）

五、有监督模型的构建

（一）数据集分割

我们使用 Python 中 sklearn 库 model_selection 模组中的 train_test_split 方法，按照 7：3 的比例将所有数据随机分为训练集和测试集。之后将训练集的特征和标签分开，对测试集也进行该操作。

（二）分类模型的合理性

由于数据集维度较高且线性不可分，我们要选取一些拟合能力强的模型。我们选取了决策树、随机森林、朴素贝叶斯分类器、logistic 回归、支持向量机五种模型[12][13]。在这里我们阐述选取它们的合理性。

1. 决策树：

（1）决策树不需要假定数据拥有特定的分布或特定的形式，其能够同时处理连续型变量和分类型变量，在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

（2）决策树非常易于理解和实现，能够直接体现数据的特点，可解释性强。如果给定一个决策树模型，那么根据所产生的决策树很容易推出相应的逻辑表达式并画出流程图。

（3）由于决策树模型较简单，易于通过静态测试来对模型进行性能表现评测。因此在进行网格搜索等超参数调参时程序运行速度非常快。

（4）决策树模型可以直接进行多分类任务，较适用于本数据集。

2. 随机森林：

（1）基学习器为决策树模型，因此随机森林也很容易实现。

（2）随机森林不需要假定数据拥有特定的分布或特定的形式。对于多种特征，随机森林可以产生高准确度的分类器，可以处理大量的输入数据。

（3）运用 Bagging 集成算法，其复杂度与使用基学习器进行训练同阶。因为决策树的训练十分快速，随机森林的训练也是一个很快速的学习过程。

（4）随机森林模型可以直接进行多分类任务，较适用于本数据集。

3. 朴素贝叶斯分类器：

（1）朴素贝叶斯算法假设了数据集属性之间是相互独立的，因此算法的逻辑性十分简单。当数据集属性之间的关系相对比较独立时，朴素贝叶斯分类算法会有更好的效果。

（2）当数据呈现不同的特点时，朴素贝叶斯的分类性能不会有太大的差异。即朴素贝叶斯算法的健壮性比较好。

（3）朴素贝叶斯算法可以直接进行多分类任务，较适用于本数据集。

4.Logistic 回归：

（1）输出值自然地落在 0 到 1 之间，有概率意义。

（2）参数代表每个特征对输出的影响，可解释性强。

（3）实施简单，计算量小。

（4）可以解决多重共线性问题。

5.支持向量机：

（1）利用核函数可以进行非线性可分数据的分类。

（2）半监督学习中也能用，两部分的支持向量机可以进行对比和借鉴。

（3）泛化能力强，在分类器中一般性能较强。

（4）支持向量机只能进行二分类问题，但可以利用一对一或一对剩余方法进行多分类问题求解。

（三）模型超参数的选取

对于我们选择的模型，需要先对数据进行 Python 中 sklearn 库 preprocessing 模组中 minmax 方法，对每列数据进行归一化。之后利用网格搜索与 k(k = 5)折交叉验证对其超参数进行选取，性能度量选择准确率和 F1 值。在附录 A 的表 A.9 中，我们给出所有模型在 sklearn 中的方法名、网格搜索的参数、网格搜索得到的最优参数。

（四）模型结果和分析

我们在这里给出五个有监督模型的交叉验证最优准确率、测试集准确率和测试集 F1 值，见表 5.4.1。

模型	交叉验证最优准确率	测试集准确率	测试集 F1 值
决策树	0.7146	0.6387	0.6742
随机森林	0.8523	0.7227	0.7873
朴素贝叶斯	0.6373	0.6229	0.6421
Logistic 回归	0.7763	0.7044	0.7702
支持向量机	0.7990	0.7023	0.7691

表 5.4.1　有监督模型的性能度量

由表 5.4.1 可见五个模型中性能最好的为随机森林，其次为支持向量机。但所有模型普遍准确率不高，我们列出随机森林与支持向量机的混淆矩阵（图 5.4.2-5.4.5），发现模型对于训练集和测试集效果基本相当，说明其泛化能力较好。因此准确率较低的原因与数据本身有关。在处理数据时我们发现，有一部分“一般困难”数据在我们（非专业人士）看起来比一部分“特别困难”的数据要困难。经过咨询辅导员和评审组得知，不同的年份、不同的学院进行评审时对标准的拿捏程度不同，因此可能会出现该情况。这可能导致一些现实评定时重要的特征在模型中的重要性被削弱了，因此导致泛化能力强但准确率并不高的现象。

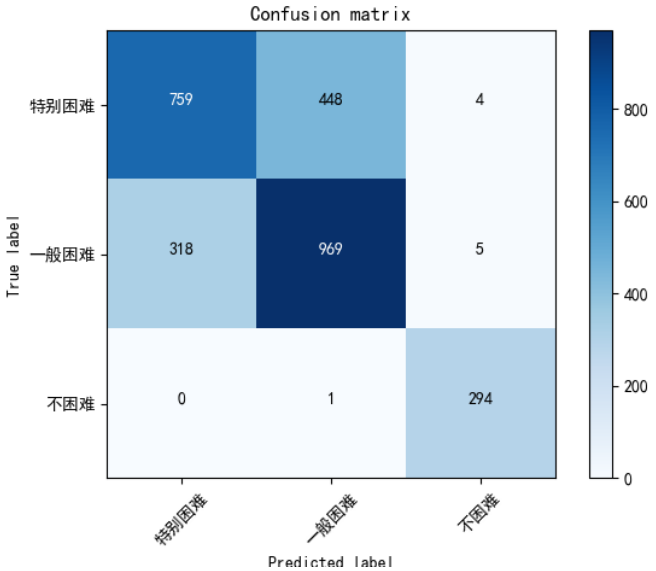

图 5.4.2　随机森林的测试集混淆矩阵

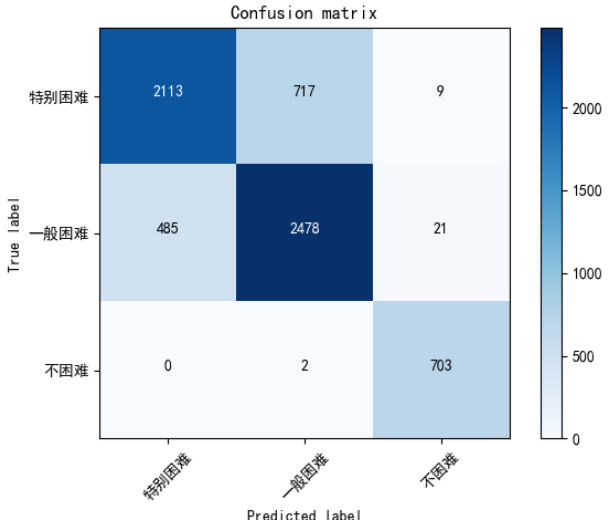

图 5.4.3　随机森林的训练集混淆矩阵

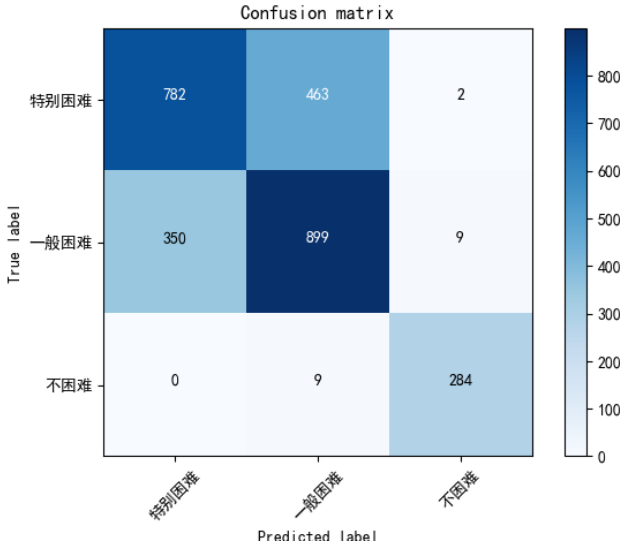

图 5.4.4　SVM 的测试集混淆矩阵

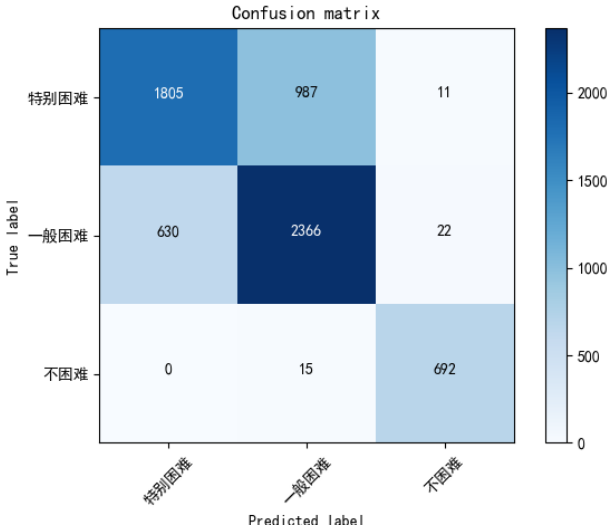

图 5.4.5　SVM 的训练集混淆矩阵

六、半监督模型的构建

（一）半监督学习的思想

我们此时考虑原数据集中被分出来并进行细化标签标注的 400 个数据，将其称为细化标签数据集（已标记样本），未被分出来的数据称为弱标签数据（未标记样本），不考虑“非困难生”数据。

若直接使用有监督学习，则仅有细化标签训练集能用于构建模型，弱标签训练集所包含的信息被浪费了；同时，由于细化标签训练集的规模较小，则由于训练样本不足，学得模型的泛化能力不强。若直接把细化标签训练集的标签删去，与弱标签训练集完全混合，进行无监督学习，则细化标签训练集所包含的信息被浪费了。我们希望利用半监督学习充分利用两部分训练集的数据，让学习器在给定少部分已标记样本（即细化标签训练集）后，不依赖外界交互便能自动地利用未标记样本（弱标签训练集）来提升学习性能。

使用半监督学习必须满足一些假设，如聚类假设，假设的本质是“相似的样本拥有相似的输出”。由数据可视化一节我们知道，本项目的数据集较好地符合聚类假设，因此可以进行半监督学习。

（二）数据集分割

我们使用 Python 中 sklearn 库 model_selection 模组中的 train_test_split 方法，按照 4：6 的比例将有标签数据随机分为有标签训练集和有标签测试集。之后将有标签训练集的特征和标签分开，对有标签测试集也进行该操作。在模型训练阶段向半监督模型提供有标签训练集特征、有标签训练集标签、所有无标签数据，在模型测试阶段提供有标签测试集特征、有标签测试集标签。实验时，我们按照弱标签将数据集一分为二，在每个数据集上单独训练半监督模型，此时问题转化为经典的有标签和无标签问题，可以使用一些现有的半监督模型。

（三）TSVM 半监督模型

半监督学习中的 SVM 统称为 S3VM(Semi-Supervised Support Vector Machine)[14]，其试图能找到将两类有标记样本分开，且穿过数据低密度区域的划

分超平面。其中最著名的是 TSVM(Transductive Support Vector Machine)[15]，其利用普通的 SVM 和伪标签思想来不断更新模型参数，直到模型收敛。

给定有标记样本集$D_l = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$，其中所有标签取值或为-1 或为 1；无标记样本集$D_u = \{x_{l+1}, x_{l+2}, \dots, x_{l+u}\}$，其中$l \ll u$。想要对无标记样本进行预测，记预测结果为$\hat{\boldsymbol{y}} = \{\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u}\}$。我们需要找到划分超平面，因此需要解决如下优化问题：

$$\min_{\boldsymbol{w},b,\hat{\boldsymbol{y}},\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C_l \sum_{i=1}^{l} \xi_i + C_u \sum_{i=l+1}^{l+u} \xi_i \tag{6.1}$$

$$s.t.\ y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, i = 1,2, \dots, l$$

$$\hat{y}_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, i = l+1, l+2, \dots, l+u$$

$$\xi_i \geq 0, i = 1,2, \dots, l+u$$

其中$(\mathbf{w}, \mathrm{b})$能确定一个划分超平面，$\boldsymbol{\xi}$为松弛向量，C_l、C_u为超参数，称为折中参数，且$C_u \ll C_l$，目的是平衡有标记样本和无标记样本的重要程度。

具体算法如下所示：

输入：有标记样本集D_l，无标记样本集D_u，折中参数C_l、C_u

输出：无标记样本的预测结果$\hat{y}$

1. 用D_l训练一个普通 SVM，记为SVM_l
2. 用SVM_l对D_u中的样本进行预测，得到暂时的预测结果$\hat{y}_l$
3. 读入C_l、C_u
4. **while** $C_u < C_l$:
5. 基于D_l，D_u，$\hat{y}_l$，C_l，C_u求解(6.1)，得到$(\mathbf{w}, \mathrm{b})$与$\boldsymbol{\xi}$
6. **while** $\exists i, j\ s.t.(\hat{y}_i \hat{y}_j < 0)$且$\xi_i > 0$且$\xi_j > 0$且$\xi_i + \xi_j > 2$:
7. $\hat{y}_i = -\hat{y}_i$
8. $\hat{y}_j = -\hat{y}_j$
9. 基于D_l，D_u，$\hat{y}_l$，C_l，C_u求解(6.1)，得到$(\mathbf{w}, \mathrm{b})$与$\boldsymbol{\xi}$
10. end while
11. $C_u = \min\{2C_u, C_l\}$

12. end while

算法在$C_u = C_l$时循环停止，第 6 行的条件为$\exists i, j$使得第 i 和 j 个数据的预测

结果不同，且其松弛向量之和过大，意味着两个标签很可能都是错误的，因此要对标签进行反号，再进行迭代。这样能够保证每次迭代后，目标函数值一定会下降，最终能保证满足循环条件，最终找到目标参数。

（四）模型结果和分析

对于 TSVM 模型，需要先对数据进行 Python 中 sklearn 库 preprocessing 模组中 minmax 方法，对每列数据进行归一化。之后利用网格搜索与 k(k = 5)折交叉验证对其超参数进行选取，性能度量选择准确率和 F1 值。在附录 A 的表 A.10 中，我们给出网格搜索的参数、网格搜索得到的最优参数。在表 6.4.1 中给出 TSVM 模型的交叉验证最优准确率、测试集准确率和测试集 F1 值。

测试集	Accuracy	F1-measure
1-2 级	0.564	0.555
3-4 级	0.623	0.621

表 6.4.1　TSVM 模型评分

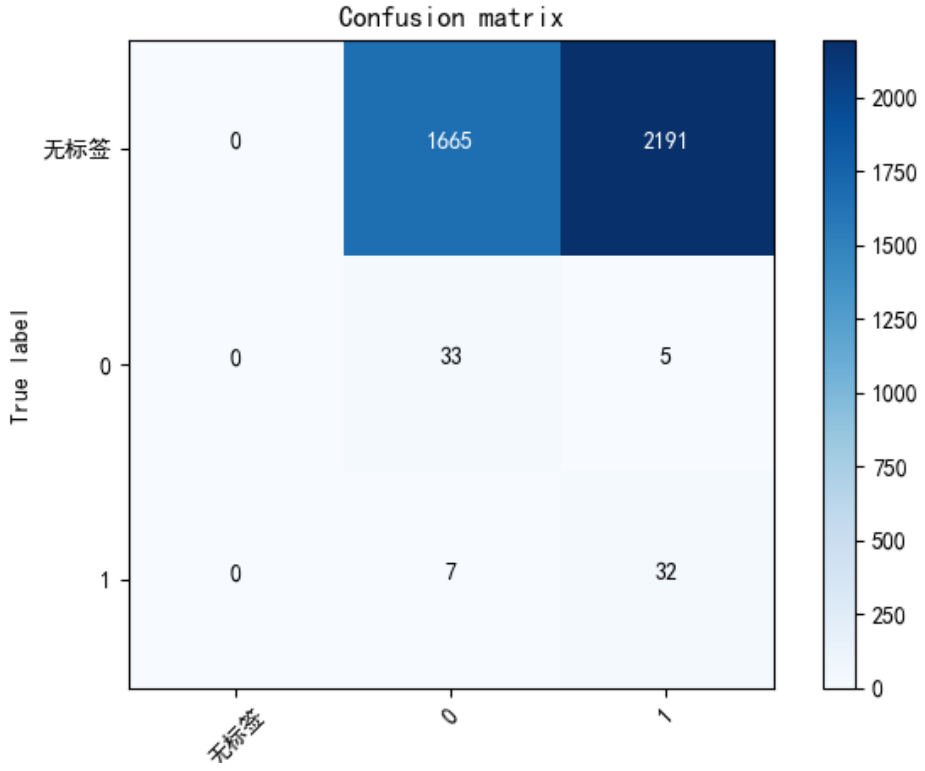

图 6.4.2　训练集混淆矩阵（包括了原本无标签的数据）

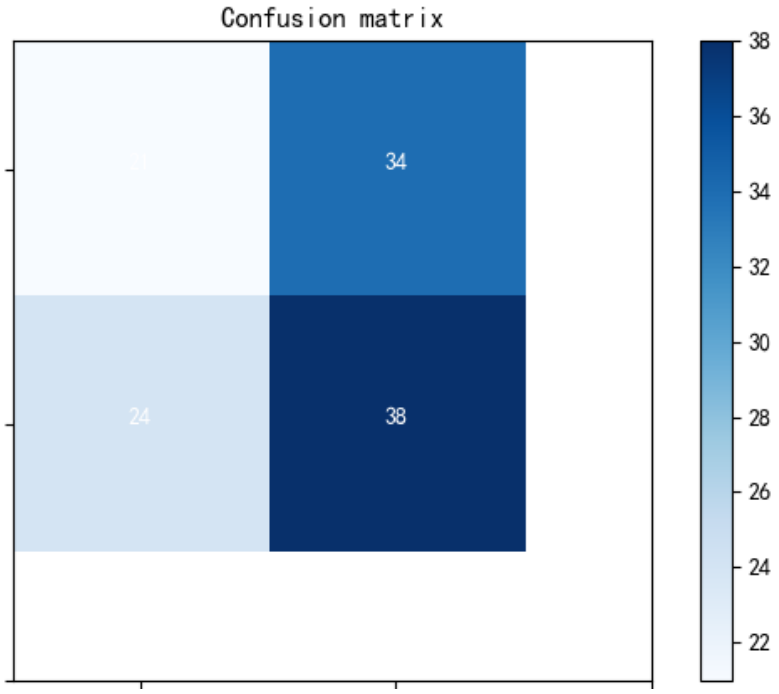

图 6.4.3　测试集

表 6.4.1 中，“1-2 级”测试集表示弱标签为“特别困难”的数据，“3-4 级”测试集表示弱标签为“一般困难”的数据，“3-4 级”的分类效果要好一些，这与我们数据降维一节的结论相符。

图 6.4.2 是在“1-2 级”训练集分类结果的混淆矩阵，注意我们这里将无标签数据也作为一类，以便于查看分类器对于无标签数据的处理结果，可以看出分类器更偏向于将样本分为 2 级，（可能需要平衡样本权重或重采样），且测试集上分类结果与训练集差距较大。

七、缺陷、总结与展望

（一）2020 年学生资助管理系统的缺陷

现有的数据是从本校往年的填写系统获取的，往年的系统对于数据的规范性并没有特别的要求，因此对数据进行处理会非常枯燥且耗时，并且可能影响数据的准确性。在 2020 年新的系统中，数据的结构化和规范性得到了大规模加强，因此我们选择将其作为特征工程的基础。但该系统还有改进的空间，系统现在所拥有的缺陷也影响了本模型预测的准确度，若能通过更改系统设计来进行优化，则会大大增加数据的准确性，进而增强模型的预测能力。现有的系统拥有包含但不限于的如下缺陷：

1. 收入来源：

在数据处理时发现有“工资”、“低保”、“退休金”、“靠哥哥姐姐工资”，“父母均有工作”、“收入来源不固定”等数据，并且这些数据并非极少数情况。而现在的自助系统上并没有这些情况的选项。

“打工”、“务农”、“做生意”实际上可以同时进行，为了解决该问题，收入来源在系统上特地注明了“若有多个经济来源，则选择主要经济来源”，但不同人对于主要经济来源的定义不同，因此该备注实际上会导致“收入来源”的影响力降低。此外，若考虑主要经济来源，则其应该作为一个单选题，但实际上设计为了一个多选题，这应该是代码编写失误。

收入来源中的“父母一方下岗”、“父母均下岗”与“打工”、“务农”、“做生意”实际上并不矛盾，很多数据均为父母没有正式工作，但在进行务农或打零工。因此应对选项进行重新设计。

2. 家庭突发情况：

在数据处理时发现有“车祸”、“家庭房屋倒塌”、“重大财产损失（如被诈骗、被盗）”、申请人自身患病或有特殊情况等数据，这些数据无法适用于现有的选项。

“父母一方无业”、“父母均无业”与“家庭主要经济来源”中的“父母一方下岗”、“父母均下岗”实际上统计的情况是一致的，“孤儿”与学生类别中的“孤残”也是一致的。但在数据处理时我们发现，经常两列数据填写有所不一致，常见的为两列中只填写一列，另一列写“无”。因此我们进行了取或运算，但即使有处理办法，不一致现象可能会导致处理后数据和该生的真实生活情况有所出入。

为了避免不一致现象，资助系统需要更改其选项。

3. 家庭总年收入：

现在的系统中的单位为(元)，但对于输入没有要求，仍可填写如“7000-11000”“三万”、“3 万”、“不多于 3 万”等数据。其应设定为必须填一个阿拉伯数字。

应让学生填写不含开支的家庭年总收入，但如何保证学生填写的数据是其不含开支家庭年总收入需要进一步讨论。

另外，有的家庭确实收入不稳定，尤其是将农作物收成作为主要经济来源的家庭。需要讨论这种情况该如何填写，使得极容易进行数据处理，又能保障学生的权利。

4. 家庭人口总数：

现在的系统对其有所备注：“三代以内”，但应令学生填写三代以内的直系赡养人数。现在的备注会导致有些学生把三代以内的旁系亲属也算进去，这样会导致出现超过 10 的异常值，此时家庭人均年收入计算也有问题。但一个问题是无法监督学生填写的是否准确，这需要进一步讨论。

另外，该项应设定为填写一个阿拉伯数字。

5. 是否贷款：

现在的系统对其有备注：“不含商业贷款”。进行数据处理时遵循的规则为尽量识别其是否为助学贷，发现有很多人在填了“是”之后描述其为商业贷款，如“是，有房贷”，因此可以推测值为“是”的数据中有一部分其实是商业贷款。因此如何让学生填写是否申请助学贷需要进一步讨论。

（二）传统评审流程的缺陷

即使系统比较严谨，我们也无法杜绝学生填写时出现谎报数据的情况；同时只通过该数据，也无法知晓该学生的平时行为习惯（如是否花钱大手大脚）。除此之外，即使这些数据是真实的，它们也无法完全概括学生的具体生活状况，只能作为其现实状况的一种描述。

要做到精准扶贫，还需要在评审时让辅导员以及同学们对参评人的状况进行描述，并令了解学生情况的专家鉴别其数据是否真实、是否准确描述其情况。因此我们的模型目前只能作为专家评审组的协助与参考，不能仅依靠模型就给出该

学生的资助方法。

（三）项目总结与展望

本项目基于本校往年的困难生数据和 2020 年网上学生资助系统，总结出了对困难生数据结构化的方法；同时给出了数据增强的一种思路，进行了数据增强。根据所得的数据建立了有监督三分类资助评级模型，对多种模型进行了性能对比，并选择了随机森林作为我们的分类模型。

此外，我们联系评定专家组对少部分数据进行了专家评定，对其给出了困难等级的细化标签。根据细化标签数据和弱标签数据，基于 TSVM 模型建立了半监督四分类自主评级模型。将有监督模型和半监督模型进行合并，则能得到最终的模型。

最后，我们对于现有的缺陷和局限性进行了分析。若资助系统和评审方法能够进行优化，我们的模型会表现得更好。

对于数据处理方法，我们未来可以尝试利用 NLP（自然语言处理）对非结构化数据进行处理，经过咨询从事 NLP 方向研究的专业人士，我们了解到 Transformer 或 Google's BERT 两个模型可能效果较好。若能利用 NLP 模型，将大大减少数据处理的时间和复杂度。

对于半监督模型，还有很多模型我们有所了解过，但由于技术和时间原因未能成功实现，如 sklearn 中的 LabelPropagation，LabelSpreading 和 SelfTrainingClassifier，QNS3VM 与 Laplacian SVM。在今后的改进中，我们可以加入更多的半监督模型来对我们的算法进行优化。

该模型在未来将用于帮助本校各个学院进行经济困难生的评审等问题。在今后的模型的改进中，我们将根据现有模型基础，通过系统的优化、数据的叠加以及信息标准化等，完成经济困难生的评审。并与该领域的专家进行模型测试与评估，将模型进一步推广和精确，可以普适地用于各学校经济困难生的认定工作。

附　录

A. 附表

特征	细化特征	变量类型
享受国家政策资助情况（多选）	建档立卡贫困户	0/1 分类变量： 0 代表“否”， 1 代表“是”
	城乡低保户	
	五保户	
享受国家政策资助情况（续）	孤残学生	0/1 分类变量： 0 代表“否”， 1 代表“是”
	军烈属或优抚子女	
家庭主要经济来源（单选）	经商	
	务农	
	退休	
	低保	
	打工	
突发事件情况（多选）	父母均下岗	
	父母一方下岗	
	祖父母患病	
	父母离异	
	父亲（母亲）患普通疾病	
	父母患普通疾病	
	兄弟姐妹患重疾	
	父亲（母亲）患重疾	
	父母患重疾	
	父亲（母亲）去世	
	突发重大自然灾害	
是否贷款（仅考虑助学贷款，不含商业贷款）		
入学前户口性质		0/1 分类变量： 0 代表“城镇”， 1 代表“农村”
民族		0/1 分类变量： 0 代表“汉族”， 1 代表“少数民族”
获得国家助学金情况		0/1/2 分类变量： 0 代表“未获得国家助学金”， 1 代表“获得二等助学金”， 2 代表“获得一等助学金”
获得助学金总金额		连续型数值变量
家庭人均年收入（单位：元）		
家庭其他成员在受教育情况	家庭成员在读大学人数	离散型数值变量
	家庭成员在读高中人数	

家庭其他成员在受教育情况	家庭成员义务教育阶段在读人数	离散型数值变量
获得助学金总个数		
家庭总人口		

表 A.1　目标特征形式

模型	变量类型
有监督三分类模型	0/1/2 分类变量： 0 代表“非困难”， 1 代表“一般困难”， 2 代表“特别困难”
半监督四分类模型	0/1/2/3/分类变量： 0 代表“困难等级 1”， 1 代表“困难等级 2”， 2 代表“困难等级 3”， 3 代表“困难等级 4”

表 A.2　目标标签形式

词性	含义	处理方法
number1	个数：1 个	视模式组中的家庭成员和学校是否合法来看保不保留。
number2	个数：2 个	
number3	个数：3 个	
number4	个数：4 个	
member	家庭成员：合法家庭成员（第一批检测，以下从略）	视模式组中的学校是否合法来看保不保留。
sp_member	家庭成员：合法家庭成员（检测并筛除第一批结果后第二批检测，以下称“第二批检测”）	
invalid_member	家庭成员：不合法家庭成员	若该词为模式组中唯一的家庭成员/学校，则与其所在的模式组全部删除不保留；否则在识别时不考虑其影响。
invalid_sp_member	家庭成员：不合法家庭成员（第二批检测）	
grad	学校：已毕业或无学历	
sp_grad	学校：已毕业或无学历（第二批检测）	
college	学校：大学及对等学历教育	视模式组中的家庭成员是否合法来看保不保留。
sp_college	学校：大学及对等学历教育（第二批检测）	
gr_college	年级：大学及对等学历教育	

high_school	学校：高中及对等学历教育	
gr_high_school	年级：高中及对等学历教育	
compulsory	学校：义务教育	视模式组中的家庭成员是否合法来看保不保留。
gr_compulsory	年级：义务教育	
others	学校：不合法的教育阶段	若该词为模式组中唯一的学校，则与其所在的模式组全部删除不保留；否则在识别时不考虑其影响。

表 A.3　“家庭其他成员在受教育情况”的词性

模式组	处理方法
学校 → 非学校或年级	所有能被处理的模式组的结束样式必为这三种中的一种，是进行模式组分组的标准。如“弟弟 高中，妹妹 初中”会被分为两组“弟弟 高中”和“妹妹 初中”。 年级可以单独出现或出现在学校之后，若出现在学校之后，则忽略此年级对模式组的影响。以下均用“学校”代指“学校”或“学校 → 年级”，“非学校”代替“非学校或年级”，因为效果是一样的。
年级 → 非学校或年级	
学校 → 年级→ 非学校或年级	
个数 → 学校 → 非学校	识别学校的等级，并在结果中对该等级学校的列加对应个数。如“1 个 大学”则在结果中“家庭成员在读大学人数”加 1。
(这里可能有个数 →) 家庭成员 → 学校→ 非学校	识别家庭成员的合法性，若不合法则直接跳过该组；否则识别学校的等级，并在结果中对该等级学校的列加对应个数（若无个数则视个数为 1)。如“2 个弟弟，大学”则在结果中“家庭成员在读大学人数”加 2。
家庭成员 → 家庭成员 →（这里可能有更多家庭成员）→学校 → 非学校	识别所有家庭成员的合法性，记录合法成员的个数，之后识别学校的等级，并在结果中对该等级学校的列加对应个数。如“大姐，二姐，大学”则在结果中“家庭成员在读大学人数”加 2。
家庭成员 → 学校 → 学校 →（这里可能有更多学校）→ 非学校	识别家庭成员的合法性，若不合法则直接跳过该组；否则识别所有学校的等级，并在结果中对该等级学校的列加 1。如“弟弟，大学，高中”则在结果中“家庭成员在读大学人数”和“家庭成员在读高中人数”加 1。

表 A.4　“家庭其他成员在受教育情况”的模式组

词性	含义	处理方法
dad	家庭成员：父亲	视模式组中的行为是否合法来看保不保留。
mom	家庭成员：母亲	
grand_parents	家庭成员：祖父母	

sp_grand_parents	家庭成员：祖父母（第二批）	视模式组中的行为是否合法来看保不保留。
siblings	家庭成员：兄弟姐妹	
sp_siblings	家庭成员：兄弟姐妹（第二批）	
invalid_member	家庭成员：不合法家庭成员	若该词为模式组中唯一的家庭成员，则与其所在的模式组全部删除不保留；否则在识别时不考虑其影响。
divorce	行为：离异	经查证，在模式组内出现该词则证明有父母离异情况。
unemployed	行为：无业	视模式组中的家庭成员是否为父母来看保不保留。
dead	行为：去世	
illness	行为：患普通疾病	视模式组中的家庭成员是否合法来看保不保留。
serious_illness	行为：患重疾	

表 A.5　“突发事件情况”的词性

模式组	处理方法
行为 → 家庭成员	所有能被处理的模式组的结束样式必为这种，是进行模式组分组的标准。如“父亲 患普通疾病 母亲 失业”会被分为两组“父亲 患普通疾病”和“母亲 失业”。
父亲 → 母亲	在原数据中查找两个词的相对位置，若父亲和母亲为两个相邻的字符/字符串，则表明“父与母”，否则为“父（母）”。
模式组里出现“祖父母”，且出现“普通疾病”或“重疾”	将“祖父母患病”设为 1
模式组里出现“离异”	将“父母离异”设为 1
模式组里出现“普通疾病”，出现“父（母）”	将“父亲（母亲）患普通疾病”设为 1
模式组里出现“普通疾病”，出现“父与母”	将“父母患普通疾病”设为 1
模式组里出现“无业”，出现“父（母）”	将“父亲（母亲）无业”设为 1，并与“家庭主要经济来源”一列中的“父母一方下岗”进行并集运算。
模式组里出现“无业”，出现“父与母”	将“父母无业”设为 1，并与“家庭主要经济来源”一列中的“父母均下岗”进行并集运算。
模式组里出现“兄弟姐妹”，且出现“重疾”	将“兄弟姐妹患重疾”设为 1
模式组里出现“重疾”，出现“父（母）”	将“父亲（母亲）患重疾”设为 1
模式组里出现“重疾”，出现“父与母”	将“父母患重疾”设为 1

模式组里出现“去世”，出现“父（母）”	将“父亲（母亲）去世”设为 1

表 A.6　“突发事件情况”的模式组

特征	(n, p)
军烈属或优抚子女	(1, 0.002)
父母均下岗	(1, 0.002)
父母一方下岗	(1, 0.02)
大学	(3, 0.01)
高中	(3, 0.01)
义务教育	(3, 0.035)
祖父母患病	(1, 0.15)
父母离异	(1, 0.01)
父亲（母亲）患普通疾病	(1, 0.01)
父母患普通疾病	(1, 0.05)
兄弟姐妹患重疾	(1, 0.003)
父亲（母亲）患重疾	(1, 0.008)
父母患重疾	(1, 0.00036)
父亲（母亲）去世	(1, 0.008)
突发重大自然灾害	(1, 0.01)
民族	(1, 0.05)
家庭人口	(7, 0.4)
入学前户口性质	(1, 0.1)

表 A.7　服从二项分布的特征的参数表

主成分序号	方差贡献率
1	0.19740356
2	0.12025587
3	0.10112684
4	0.09057711
5	0.06853096
6	0.06091598
7	0.05509322
8	0.05110082
9	0.04114201
10	0.03504193

表 A.8　前 10 个主成分的方差贡献率

模型	网格搜索初始参数	网格搜索得到最优参数
决策树	'dt__max_features': ['auto', 'sqrt', 'log2'], 'dt__class_weight': [None, 'balanced'], 'dt__ccp_alpha': [0.0, 0.1], 'dt__min_impurity_decrease': [0., 0.01], 'dt__min_samples_leaf': [1, 5], 'dt__min_samples_split': [2, 8],	'estimator__dt__max_features': None, 'estimator__dt__class_weight': None, 'estimator__dt__ccp_alpha': 0.0, 'estimator__dt__min_impurity_decrease': 0.0, 'estimator__dt__min_samples_leaf': 1, 'estimator__dt__min_samples_split': 2,
随机森林	'rf__criterion': ['gini', 'entropy'], 'rf__n_estimators': [100, 300, 600, 800, 1200], 'rf__min_samples_split': [2, 5], 'rf__min_samples_leaf': [1, 4], 'rf__bootstrap': [True, False], 'rf__min_impurity_decrease': [0., 0.01, 0.1], 'rf__class_weight': ['balanced', 'balanced_subsample', None], 'rf__warm_start': [True, False], 'rf__oob_score': [True, False], 'rf__ccp_alpha': [0., 0.1, 0.5]	'estimator__rf__criterion': 'gini', 'estimator__rf__n_estimators': 100, 'estimator__rf__min_samples_split': 2, 'estimator__rf__min_weight_fraction_leaf': 0.0, 'estimator__rf__bootstrap': True, 'estimator__rf__min_impurity_decrease': 0.0, 'estimator__rf__class_weight': None, 'estimator__rf__warm_start': False, 'estimator__rf__oob_score': False, 'estimator__rf__ccp_alpha': 0.0,
朴素贝叶斯	'NB__var_smoothing': [1e-10, 1e-9, 1e-8, 1e-6, 1e-4, 1e-2, 1],	'estimator__NB__var_smoothing': 1e-09
Logistic 回归	'Logistic__penalty': ['l1', 'l2', 'elasticnet', 'none'], 'Logistic__C': [0.0001, 0.001, 0.01, 0.1, 1, 2, 5, 10, 100, 1000], 'Logistic__solver': ['lbfgs', 'liblinear', 'newton-cg', 'sag', 'saga'], 'Logistic__fit_intercept': [True, False], 'Logistic__dual': [True, False], 'l1_ratio': [True, False], 'warm_start': [True, False], 'intercept_scaling': [0.01, 0.1, 0.5, 1, 2, 5, 10]	
支持向量机	'SVM__kernel': ['linear', 'rbf', 'poly'], 'SVM__C': [0.7, 0.8, 0.9, 0.95, 1, 1.05, 1.1, 1.2, 1.5, 2], 'SVM__degree': [2, 3, 4],	'estimator__SVM__kernel': 'rbf', 'estimator__SVM__C': 1.0, 'estimator__SVM__degree': 3, 'estimator__SVM__decision_function_shape': 'ovr',

	'SVM__decision_function_shape': ['ovo', 'ovr'], 'SVM__break_ties': [True, False],	'estimator__SVM__break_ties': False,

表 A.9　有监督学习网格搜索初始参数与结果

B. 源码仓库及文档

1. 源代码仓库地址（包含源码、模型、数据样例、安装文档等）：
   https://github.com/Alexhaoge/MLSR
2. API 文档：https://www.alexhaoge.xyz/mlsr/index.html

参考文献

[1]. 全国学生资助管理中心.2019 中国学生资助发展报告[N]. 人民日报，2020-05-21，7

[2]. 陆孙琦.高校助学金评选发放所面临的问题与对策[J]. 轻工科技，2013，5：152-153.

[3]. 张善红.国家助学金评定中的量化模式研究[J]. 高教学刊，2015，15：101-102.

[4]. 刘美荣.我国高校助学金使用的现状分析[J]. 中国管理信息化，2014，4：141-142.

[5]. 李亚员.“精准资助”原则指导下的高校学生资助模式创新[EB/OL]. http://www.xszz.cee.edu.cn/index.php/shows/22/2600.html. 2016-07-22

[6]. 欧阳铁磊. 叶玲肖. 基于大数据分析的高校贫困生精准资助策略研究[J]. 计算机应用与软件. 2020, 8:45-47.

[7]. 结巴中文分词[CP／DK]. https://github.com/fxsjy/jieba. 2020-01-20

[8]. Pedregosa *et al.* (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12. 2825-2830.

[9]. Buitinck *et al.* (2013). API design for machine learning software: experiences from the scikit-learn project. *arXiv:1309.0238 [cs.LG]*

[10]. Hinton, G.E., Roweis S.T. (2002). Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems,* volume 15, pages 833–840, Cambridge, MA, USA. The MIT Press.

[11]. Laurens, M., Geoffrey, H. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579-2605.

[12]. 周志华. 机器学习. 北京：清华大学出版社，2016. 57-66, 73-95, 121-144, 147-169, 172-181

[13]. 李航. 统计学习方法（第二版）. 北京：清华大学出版社，2019. 59-66, 67-89, 91-94, 111-154

[14]. 周志华. 机器学习. 北京：清华大学出版社，2016. 298-300

[15]. Joachims, T. (2001). Transductive Inference for Text Classification Using Support Vector Machines. ICML.

致　谢

学工部学生资助中心提供了本项目所用的数据，并给出了数据结构化的目标。数学科学学院的宫老师对非结构化数据的处理方式提供了丰富的指导，并且其和专家组其他老师进行了数据细化标签的处理。统计与数据科学学院的徐老师对家庭收入的分布构造提供了方法与指导。数学科学学院 2017 级本科生范同学、代同学、程同学提出了特征工程中模式组识别的算法，并对于其中关键词的选取提供了建议。此外，外校的范同学建议我们在模式组识别中使用分词的方法和 jieba 库。

向以上所有对本项目提供帮助的老师和同学们表示我们最诚挚的谢意！