银行精准营销解决方案

武笑石
2016011595
软件 73

摘要

在这次作业中，我利用 scikit-learn 所提供的框架针对银行精准营销解决方案的任务进行了实验。实验中，我用到了多种不同的分类方法、多种特征选择、以及多种模型评价方法。此外，我自己在 scikit-learn 所提供的模板基础上手动实现了逻辑回归模型，并分析了其与 scikit-learn 库中的实现的异同。

1. 实验报告结构

这份实验报告将分为七部分进行介绍。第二部分将交代实验环境，第三部分交代了实验代码的文件结构以及使用方法。第四、五、六部分按照数据预处理、模型、评估的结构分别进行介绍。第七部分展示并分析了实验结果。

2. 实验环境说明

操作系统	Windows 10 education
CPU	Intel i5-8250U
python 版本	3.6.10
虚拟环境	conda
关键依赖	numpy=1.18.4 pandas=1.0.3 scikit-learn=0.23.1

3. 代码结构说明

3.1 文件结构

task1.py	程序的入口，数据预处理、模型搭建、模型评估逻辑均在此文件 中实现。代码中有完整清晰的注释。
logistic_regression.py	自己实现的逻辑回归分类器。我的实现利用到了 numpy, math。该类实现了 fit, predict 接口，以嵌入到主程序的流水线中，具体的算法逻辑上完全由我手工实现。代码中同样有完整清晰的注释。
data/	训练数据，为确保程序正常运行，请务必将 data 文件夹置于上述两个代码文件的同级目录。

3.2 使用方法说明

本报告中的所有实验共享同一个入口，也就是执行 task1.py 文件。可以通过配置不同参数的方式改变实验条件。以下表格介绍各命令行参数的含义。

此外，可以直接执行:

python task1.py –help

以查看更详细的使用说明。

命令行参数	意义	取值范围
--drop_feature_list	一个列表，该列表中出现的 feature 将不会在训练中被使用。	数据集中出现的类别。
--classifier	指定使用的分类器。自己手工实现的分类器同样可以在这里指定。	“logistic”, “random_forest”, “mlp”, “svm”, “mylogistic”
--folds	K-Fold 参数。	任何大于 1 的整

		数
--max_iter	solver 最大迭代次数。对 random forest 不适用。	正整数

4. 数据预处理

此次作业中用到的数据集中所提供的特征中包含多个枚举类型，难以直接用于训练；并且其他数值类型的参数也具有不同的分布，不利于训练。因此有必要进行数据预处理。

数据预处理包含三部分：

1. 选择 feature。
2. 将枚举类特征编码成 one-hot vector。
3. 将数值类型特征进行归一化。

选择 feature 过程按照提供的命令行参数进行处理，支持手动选择。

枚举类特征编码的过程使用到了 scikit-learn 中的 OneHotEncoder 类，一个具有 n 种可能取值的枚举类将被编码成为 n 个 binary 的类别。

特征归一化过程中用到了 scikit-learn 中的 StandardScaler 类，将所有数值类型的变量将被线性压缩到 0-1 之间。

此外，值得一提的是，这个数据集明显是一个不平衡的数据集，真样例的数量远远小于假样本的数量，这也是后续实验分析中需要考虑的因素。

5. 模型

此次作业中对五种不同的模型进行了测试。分别为：逻辑回归模型、随机森林、多层感知机、支持向量机以及自己手工实现的逻辑回归模型。

特别强调说明的是，所有迭代逻辑、参数更新逻辑完全由我手工实现。虽然用到了 numpy 库，但只是为了简化数学运算，所有的训练逻辑、参数更新逻辑均由我手工实现。

在实验中，我发现我实现的逻辑回归与 scikit-learn 中提供的版本之间在验证集有较大的性能差距，但在训练集上差距较小，因此初步判定为过拟合导致的泛化性能下降，因此我增加了 L2 正则项，增加后虽然性能没有达到 scikit-learn，但是已经取得了很大的提高。相关内容将在第七部分实验结果部分进行进一步更详细的说明。

6. 评估方法介绍

本次实验按照作业说明的要求进行了 K-Fold 交叉验证。具体 K 为多少可以在命令行参数中进行指定。

模型结果评估用到了 scikit-learn 中的 classification_report 类，该类自动生成一个结果评估报表，其中包含了正样本、负样本的准确率、召回率、F 值，以及支持样本数量。此外还有同时考虑正负样本后得到的相关指标。

在模型评估时，我主要选择真样本对应的召回率与精确度，原因如下。

首先，在这个具体任务中，该模型对于银行的价值在于找出潜在客户，并进行精准营销。根据社会常识，谈成一笔交易的所带来的收益远远高于一次失败的宣传中所付出的成本，因此我们对模型的期待应当是在找出绝大多数潜在客户的前提下，再尽可能排除掉显然不可能达成交易的客户，以减少宣传成本、避免打扰客户。

如果用机器学习的语言来描述，优化目标是在确保真样本的召回率的前提下，尽可能提高真样本的精度。之所以不选用更加简洁的 F 值，是因为在 F 值的计算过程中，精度与召回率在数学上是等价的，而在这个任务中，根据上述分析，召回率的重要性高于精度。

7. 实验结果

这部分中，我将在 7.1 节对比不同模型的效果，并分析差异原因。在对比不同模型时，我使用了除了用户编号以外的全部特征。

此外，我将在 7.2 节对比不同数据输入特征选择下，模型的运行效果。这时我将使用上述分析中效果较好、且适合比较的模型。

在所有实验中，均使用 10 折交叉验证。除了多层感知机以外，其他模型的最大迭代次数均设置为 1000 次。出于效率考量，多层感知机以及我实现的逻辑回归的最大迭代次数设置在了 100 次。

7.1 分类模型对比

模型	真样本召回率	真样本精度	真样本 F 值（不比较，仅参考）	判定为真的结果比例
逻辑回归	35%	66%	0.46	7%
随机森林	39%	66%	0.49	8%
多层感知机	50%	61%	0.55	9%
支持向量机	58%	17%	0.26	40%
逻辑回归（自己手写）	52%	48%	0.50	12%

首先对表头做一个解释。由于第六部分中解释的原因，比较模型时，最先衡量的量时正样本召回率，其次是真样本精度。F 值也是重要的模型衡量指标，因此在表格中给出，但不做参考。表格中同样给出了所有样本中被模型判定为真样本的比例，该值在此问题的模型评价中并没有直接的物理意义，但是可以很直观地反映模型在判定上的策略与行为。

如果只看真样本召回率，那么支持向量机无疑是最高的，但是高召回率的代价是非常低的精度。从判定为真的比例可以看出，支持向量机的策略是将大量样例判断为真，来提高召回率，因此召回率高并不能说明模型的判别能力强。

如果从理论上来分析，支持向量机是一个线性分类模型，而且是以高维平面作为决策边界的简单线性模型，表达能力自然有限。这个问题显然是非线性的，例如，5 岁的小孩通常不会、也没有权限参与银行的业务，而大多数老年人在商业上并不活跃，与银行的关系也较为简单。因此年龄与办理银行业务之间的关系很可能是一个钟形曲线，而这是线性分类器无法表达的逻辑。

除去支持向量机，余下的模型中多层感知机的效果是最好的（我实现的做了少量调参，一同比较不公平）。多层感知机是非线性模型，可以很好地处理非线性问题。而且多层感知机参数量远远高于其他模型，因此拟合能力也远强于其他模型。但相应的，参数量大在理论上会导致泛化能力的下降，但很幸运的是从结果来看多层感知机的泛化能力依然在其他模型之上。多层感知机美中不足的是收敛时间远远高于其他模型。

从真样本召回率来看，逻辑回归的效果是最差的，但是我实现的版本效果可以与多层感知机媲美，其原因如下。除了加正则项、weight clipping 等常规操作外，我修改了逻辑回归的决策边界。注意到这个数据集存在真假样例不平衡的问题，这也就意味着整体上的参数更新更加倾向于假样例，存在系统误差。因此将决策边界从 0.5 下调理论上会有很大帮助。经过实验，我发现将此参数设置在 0.2 时模型效果最佳，可获得表中结果。

7.2 不同输入特征对模型效果的影响

这一部分的实验选用我自己实现的分类器进行对比。首先下表是实验结果。

输入特征	真样本召回率	真样本精度	真样本 F 值（不比较，仅参考）	判定为真的结果比例
全部	52%	48%	0.50	12%
部分	50%	47%	0.49	12%
全部+ID	99%	59%	0.74	20%

我尝试了多种组合，但是很遗憾，大部分组合的实验结果都是与用全部特征相比略有下降，即便有一些性能提升，也都有明显的代价（例如召回率可以达到 70%，但精度下降明显）。上表中的实验组是全部特征排除 day 以及 month 两个特征后得到的结果，以作为参考。之所以排除这两个特征是因为经验上讲最后一次联系的时间和结果的相关性可能并没有那样明显。

我个人认为，在模型训练的过程中，短板效应十分明显，如果想要取得性能提升，最有效的思路并不是锦上添花，而是找出系统短板，针对痛点下手。从正负样例悬殊的准确率可以猜测，这个任务中系统的痛点可能在于真假样例的极度不均衡，因此解决这一问题或许是取得性能提升的关键所在，特征选择可能在其他很多任务中能够起到提高泛化性能的作用，但在此次实验中确实并没有能够得到充分体现。

虽然如此，这个数据集上有反例可以证明输入一个不合理的特征对分类结果会产生多大的影响。正如前文提到，我在进行上述所有实验时都排除了用户 ID 这一特征，但如果加入这个特征，得到的结果会出人意料得好。然而 ID 本身只是客户编号，本身并不包含任何描述客户属性的信息，理应与分类无关。那么答案只有一个了，ID 与标签之间有人为的强耦合关系，检查数据后的确也可以证实这一点，编号为 22357 及之后的所有数据标签均为真，数据的标签恰好以此为分界线。

这一反例也说明了训练前仔细检查数据的重要性。