机器学习导论 习题六

211300063, 张运吉, 211300063@smail.nju.edu.cn

2023 年 6 月 25 日

作业提交注意事项

1. 请在 LaTeX 模板中第一页填写个人的学号、姓名、邮箱;
2. 本次作业需提交作答且运行后的 .ipynb 文件与相应的 .pdf 文件; **请打包为 .zip 文件上传**. 注意命名规则, 文件均命名为“学号_姓名”+“. 后缀”(例如 211300001_张三”+ “.ipynb”、“.zip”);
3. 若多次提交作业, 则在命名 .zip 文件时加上版本号, 例如“211300001_张三_v1.zip”(批改时以版本号最高的文件为准);
4. 本次作业提交截止时间为 **6 月 25 日 23:59:59**. 未按照要求提交作业, 提交作业格式不正确, **作业命名不规范**, 将会被扣除部分作业分数; 除特殊原因 (如因病缓交, 需出示医院假条) 逾期未交作业, 本次作业记 0 分; **如发现抄袭, 抄袭和被抄袭双方成绩全部取消**;
5. 本次作业提交地址为 here, 请大家预留时间提前上交, 以防在临近截止日期时, 因网络等原因无法按时提交作业.

Overview: Final Homework

本次作业将从一个医疗数据集 PPG-DaLiA 入手, 对本门课程进行系统性地回顾与考察.

首先对 PPG-DaLiA 数据集进行介绍. 该数据集的构建目的为心率估计, 包含 15 个佩戴生理和运动传感器的受试者在进行各类活动时的监测数据. 数据集已预先进行了处理与划分, 其中训练集包含 466,160 个样本, 测试集包含 51,796 个样本. 每个样本有 19 个特征:

- chest_*, wrist_* 以及 rpeaks 这 11 个特征为传感器实时数据, 具体含义可见数据集说明文档, 此处不再赘述;
- gender, age, height, weight, skin, sport 这 6 个特征为受试者的个人信息, 依次为性别、年龄、身高、体重、肤色类别以及锻炼频率, 其中 sport 越大代表运动越频繁;
- activity 为受试者当前的活动类型, 分为 9 个类别, 可作为分类任务的标签. 具体地, 该特征从 0 至 9 依次为不同活动间的切换、站立、上下楼梯、桌上足球、骑自行车、开车、午休、走路以及工作;
- heart_rate 为受试者当前心率, 可作为回归任务的预测目标.

由特征语义可知, PPG-DaLiA 数据集既可构建分类任务, 也可构建回归任务. 因此本次作业将分为两大部分, 其中前四题将通过分类任务, 完成对数据集的熟悉以及对课程知识的回顾; 第五题则通过回归任务, 完成对所学知识的实践.

本次作业具体组织如下:

- 第一题 [10pts]: 对数据集进行分析, 并做相应处理;
- 第二题 [15pts]: 以逻辑回归为例, 在不同的性能度量下 (Accuracy 与 AUC), 通过 K 折交叉验证选取线性模型的超参数;
- 第三题 [15pts]: 实现其它课程中学过的算法, 包括决策树、多层感知机、支持向量机、朴素贝叶斯、随机森林 (Bagging 集成算法) 以及 LightGBM (Boosting 集成算法);
- 第四题 [15pts]: 对训练得到的各类学习器进行结合, 分别实现 Voting (投票法) 与 Stacking (学习法) 两种结合策略;
- 第五题 [45pts]: 基于前四题内容, 构建模型完成回归任务 (不限制方式, 自由发挥), 并对实现过程进行说明. 模型评估将通过 Kaggle 平台完成.

本次作业需提交的文件:

- hw6.ipynb: 包含五道题相应的代码与运行记录;
- hw6.pdf: 在本文档的基础上, 包含对第四题与第五题相应内容的说明.

1 [10pts] Dataset Analysis and Preprocessing

当面对一个陌生的机器学习任务时, 首先需要做的是对数据集进行分析, 包括数据集大小、数据分布、特征类型以及缺失情况等. 分析结束后, 再针对具体情况对数据集进行相应处理.

(1) **[2pts]** 输出训练集和测试集每一维特征的名称与类型, 并找出测试集缺失的一维特征, 完成 hw6.ipynb 中 1.1 内容.

(2) **[2pts]** 对每一维特征进行分析并完成 hw6.ipynb 中 1.2 内容:

- 数值型特征 (float, int) 统计最小/大值、均值、标准差、去重后的元素个数;
- 类别型特征 (object) 统计去重后的元素个数, 并输出不同取值对应的样本数;
- 每一维特征统计缺失数量.

(3) **[2pts]** 对于 object 型特征, 通常需要对其进行编码, 将其转为数值型特征, 便于后续模型训练. 常用的编码方式有两种: 定义映射函数进行转换与独热编码 (One-Hot). 此处我们以第一种为例, 将所有 object 型特征从 0 开始编码, 例如某个特征有三个取值, 则映射为 0, 1, 2. 需保持训练/测试集映射函数一致, 并完成 hw6.ipynb 中 1.3 内容.

(4) **[2pts]** 在训练集上, 分析任务标签的数据分布, 并完成 hw6.ipynb 中 1.4 内容:

- 画出 activity 特征 (分类任务标签) 的数据分布柱状图, 其中横坐标为特征取值, 纵坐标为该取值对应的样本数;
- 画出 heart_rate 特征 (回归任务标签) 的数据分布直方图, 即先将取值范围均匀划分为 30 个等宽区间, 再统计各区间内样本数.

(5) **[2pts]** 为了对数据集各特征之间的关系有进一步的认识, 通常会采用一些统计量来度量特征两两之间的相关性. 此处以 Pearson 相关系数为例, 要求计算出特征两两之间的相关系数, 并通过热力图进行展示. 具体内容见 hw6.ipynb 中 1.5 部分.

2 [15pts] K-Fold Cross Validation

交叉验证法是机器学习中常用的模型评估方法. 请仔细阅读学习课本第二章 2.2.2 节, 并在本题中使用 K 折交叉验证来为带有正则项的对数几率回归选取合适的超参数, 本题中涉及的超参数为正则化系数, 其定义及意义参见课本第六章 6.4 节.

(1) **[6pts]** 使用精度作为性能度量, 通过 K 折交叉验证法来为对数几率回归选择合适的超参数, 完成并运行 hw6.ipynb 中 2.1 内容.

(2) **[2pts]** 正确运行 hw6.ipynb 中 2.2 代码块, 绘制出结果, 并根据结果选择一个合适的超参数, 将其填写在 2.2 代码块的下方.

(3) **[5pts]** 使用 AUC 作为性能度量 (ovr), 通过 K 折交叉验证法来为对数几率回归选择合适的超参数, 完成并运行 hw6.ipynb 中 2.3 内容.

(4) **[2pts]** 正确运行 hw6.ipynb 中 2.4 代码块, 绘制出结果, 并根据结果选择一个合适的超参数, 将其填写在 2.4 代码块的下方.

3 [15pts] Various Classification Models

分类问题是机器学习中常见的一类问题, 本题中将带大家实现课程上涉及到的分类模型. 请阅读学习课本上的对应章节, 并严格按照要求在 hw6.ipynb 的代码块中完成相应任务.

(1) **[2pts]** 调用 sklearn 库实现决策树模型, 完成并运行 hw6.ipynb 中 3.1 内容.

(2) **[2pts]** 调用 torch 库实现多层前馈神经网络模型, 完成并运行 hw6.ipynb 中 3.2 内容.

(3) **[2pts]** 调用 sklearn 库实现支持向量机模型, 完成并运行 hw6.ipynb 中 3.3 内容.

(4) **[2pts]** 调用 sklearn 库实现朴素贝叶斯模型, 完成并运行 hw6.ipynb 中 3.4 内容.

(5) **[2pts]** 调用 sklearn 库实现随机森林模型, 完成并运行 hw6.ipynb 中 3.5 内容.

(6) **[2pts]** 调用 sklearn 库实现 LightGBM 模型, 完成并运行 hw6.ipynb 中 3.6 内容.

(7) **[3pts]** 完成并运行 hw6.ipynb 中 3.7 内容, 将上述模型使用 ovr 的 ROC 曲线绘制在同一张图内. 根据结果分析在该问题上哪些模型效果较好, 哪些模型效果较差, 将其填写在 3.7 代码块下方.

在第三次与第四次作业中, 已实现过其中的一些模型, 代码可以直接使用. 需要参考的库有:

- Scikit-learn: https://scikit-learn.org/stable/index.html;
- PyTorch: https://pytorch.org/docs/stable/index.html.

4 [15pts] Model Combination Strategies

为在给定的学习任务上取得更好的性能, 一种常见的方式是将多个模型进行结合, 使结合后的模型比单个模型表现地更好. 模型结合策略有很多, 例如平均法、投票法以及学习法. 在实践中通常尝试多种结合策略, 以期取得更好的效果. 更多内容可见课本第八章 8.4 节.

(1) **[5pts]** 投票法是分类任务上最为常见的结合策略. 此处要求对先前得到的七个不同模型, 采用相对多数投票法的策略进行结合, 完成并运行 hw6.ipynb 中 4.1 内容.

(2) **[5pts]** 当训练数据很多时, 通常可以采用学习法对模型进行结合. Stacking 是学习法的典型代表 (详细内容见课本第八章 8.4.3 节), 此处要求采用 5 折交叉验证的方式实现该种结合策略. 具体地, 初级学习算法采用多层感知机、支持向量机与 LightGBM 三种, 次级学习算法采用逻辑回归. 完成并运行 hw6.ipynb 中 4.2 内容.

(3) **[5pts]** 简述上述两种结合策略的实现过程, 并对两种策略的运行结果进行简要分析.

Solution. 此处用于填写第三小问的回答 (中英文均可)

相对多数投票法：

第三题已经获得了七种模型的预测结果，我把它们平成一个 numpy 矩阵，其中第 i 行代表七个模型对样本 x_i 的预测结果，然后只需要统计每一行出现最多的数字就是最终的预测结果.

Stacking 法:

采用 5 折交叉验证，先把原始数据集 X_train 分成五份, 然后在每一折选择四份作为训练集训练基学习器，然后让基学习器对剩余一份预测，把预测结果记录下来，作为次级学习算法的训练集. 5 折交叉验证结束后，用刚刚基学习器的预测作为训练集训练次级分类器.

运行结果:

相对多数投票法效果更好. 相对多数投票法简单且易于实现，不需要额外的模型训练和特征工程. Stacking 的实现相对复杂，需要进行交叉验证来训练和预测多层模型，需要更多的计算资源和时间. 我认为，相对多数投票法运行结果更好的原因是这个模型并不是很复杂，数据量也不是很大，使用投票法已经足够有效，而 Stacking 法需要在数据量比较多的时候才会显示出优势.

5 [45pts] Regression Task in Practice

回归问题是机器学习中的一类重要问题. 本题需要基于前四题内容, 构建模型完成对数据集中 heart_rate 进行预测的回归任务. Kaggle 平台 (https://www.kaggle.com) 是一个常用的机器学习竞赛平台, 本题涉及到的数据、评分等将统一在该平台上实现.

请在 Kaggle 平台完成注册, 并点击该链接进入比赛界面. 请注意:

- **带标记的训练数据集与不带标记的测试数据集均在比赛中的 Data 栏目下载, 与作业 zip 包中自带的数据集一致.**
- **点击比赛右上角 Submit Predictions 进行预测结果上传, 每天最多上传 3 次.**
- **上传的预测结果为带表头的双列 csv 文件;**

  其中第一列表头为 id, 每行的值为 0, 1, ···; 0 对应于测试数据中第 1 行样本, 以此类推. 第二列表头为 expected, 每行的值为对测试数据中对应行样本的预测. 提交文件的格式示例请参考 data 文件夹下的 submission_eg.csv 文件.
- **请将队名更改为“学号”后再上传预测结果, 例如“211300001”, 否则不计分.**
- **可以在比赛中的 LeaderBoard 栏目查看自己最新的提交与得分.**

具体的给分细则如下所示:

- **[5pts]** 将实现代码填入 hw6.ipynb 的相应位置并运行.
- **[5pts]** 模型在测试集上预测结果的 RMSE 小于 baseline 1 (23.1).
- **[15pts]** 在 hw6.pdf 中给出具体实现过程的分析与说明, 涉及模型的实现方法、过程中遇到的难点以及相应的解决措施等.
- **[20pts]** 在预测结果的 RMSE 小于 baseline 2 (20.7) 的前提下, 进行如下评分:

$$\text{score} = 12 + 8 \times \left(1 - \frac{\text{你的排名} - 1}{\text{达到 baseline 的总人数}}\right)$$

Solution. 此处用于填写具体实现过程的分析与说明 (中英文均可)
首先要获取训练集，因为和前四题的分类问题不一样，需要重新获取，就是把 train_data 中的前 18 维获取作为 X_train，最后一维”heart_rate” 作为 y_train.
然后要选择一个合适的模型，这里根据前四题的结果大致可以发现使用一些简单的线性模型 (比如 svm) 效果不是很好，因为可能根本找不到相关的划分超平面也不会设置合适的核函数，最终我选择使用集成学习来做这道题. 我尝试了随机森林和 Adaboost 模型，经过对比发现随机森林比较好，于是我最终选用随机森林.
还有就是是否要对数据进行 normalize，这里对比发现是不 normalize 效果更好，原因我觉得是因为在随机森林中，每个决策树都是根据特征的相对顺序来进行节点划分的，而不考虑特征的绝对值.
最后就是超参数的选择，一开始我把随机森林决策树的数量设置为 10，发现效果并不好，应

该是因为决策树数量太少，训练效果不好欠拟合了，所以我把决策树的数量调高.