Lab5 实验报告

一、实验目的：

综合本课程所学习知识，用线性回归模型、决策树模型、神经网络模型、支持向量机以及 XGBoost 等分类模型来完成标签预测任务，并分析实验结果

二、实验原理：

1.线性回归模型：对于分类任务，在广义线性回归模型中，用 sigmoid 函数将线性回归模型的预测值与分类标签关联起来，即可得到对数几率回归（logistic regression）。
对于二分类任务，预测样本$\boldsymbol{x}$为正例的可能性为

$$y = \frac{1}{1 + e^{-(\boldsymbol{w}\,x+b)}}$$

当$y \geq 0.5$时将其划分为正例，否则为反例。
通过最大化对数似然

$$l(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln\, p(y_i|\boldsymbol{x}_i; \boldsymbol{w}, b)$$

来确定模型中的参数$\boldsymbol{w}, b$。为了减少过拟合风险，常在上式中添加正则项，常用的为 L1 和 L2 正则项（C为超参数，控制正则项强度）

$$f_1(\boldsymbol{w}, b) = -\sum_{i=1}^{m} \ln\, p(y_i|\boldsymbol{x}_i; \boldsymbol{w}, b) + \frac{1}{C}(\|w\|_2^2 + b^2)$$

$$f_2(\boldsymbol{w}, b) = -\sum_{i=1}^{m} \ln\, p(y_i|\boldsymbol{x}_i; \boldsymbol{w}, b) + \frac{1}{C}(\|w\|_1 + |b|)$$

2.决策树模型：对于分类任务，在决策树模型中，从根节点出发，在内部节点对特定属性进行测试，导出判定结果或进一步的判定问题（进入新的节点）。从根结点出发到每个叶结点的路径对于了一个判定测试序列。决策树的训练方式遵循简单的“分而治之”策略，每次划分的方式（属性和阈值）用信息熵（information entropy）增益或者划分后基尼系数（gini）来决定。该模型防止过拟合的主要手段是剪枝。

3.神经网络模型：多层感知机（MLP）是神经网络模型的一种，由多层“神经单元”组成，具有非常强大的拟合能力。神经网络常遭遇过拟合问题，通常用“早停”策略或添加正则项等方式解决。可调整参数：感知机激活函数（一般为：单位映射、sigmoid 函数、Relu 函数等）、网络层数以及每层神经元数、正则项大小、是否早停等

4.支持向量机：在分类任务中，该模型试图在数据空间中寻找一个将不同类型划分开的超平面，且使得两类数据距超平面的距离最大。训练该模型，即求解以下最优化问题：

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

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

可以转化为对偶问题用 SMO、梯度下降等方法求解。

5.XGBoost:XGBoost 是由多个基模型组成的一个加法模型,假设第k个基本模型是 $f_k(x)$,那么前 t 个模型组成的模型的输出为

$$\hat{y}^{(t)} = \sum_{k=1}^{t} f_k(x) = \hat{y}^{(t-1)} + f_t(x)$$

在学习第 t 个基模型时，XGBoost 要优化的目标函数为:

$$Obj^{(t)} = loss\left(\hat{y}^{(t)}, D\right) + \sum_{k=1}^{t} penalty(f_k)$$

$$= loss\left(\hat{y}^{(t-1)} + f_t(x), D\right) + penalty(f_t)$$

其中 loss 为基学习器要最小化的误差函数，penalty 为 XGBoost 模型的复杂度惩罚项。对比基学习器的优化公式，只需给第 t 个基决策树投放如下数据进行训练：

$$D^{(t)} = \left\{\left(\boldsymbol{x}, y - y^{(t-1)}\right) \middle| (\boldsymbol{x}, y) \in D\right\}$$

和神经网络一样，XGBoost 模型的拟合能力很强，需要注意防止过拟合。

6. 用二分类学习器完成多分类任务：常用手段为一对一（OvO）、一对多（OvR）、多对多（MvM）等。本次采用 OvO 模型，为了避免接口出错可能导致的问题，直接调用 sklearn 库的实现。

7. 填补缺失数据：常用方法有：重采样法、用平均值（中位数）填补、K 近邻方法填补（KNN）、直接过滤等

三、实验步骤：

0. 文件功能划分
   main.ipynb：实验记录
   Preprocess.ipynb：预处理记录，记录了数据预处理的草稿
   funs.py：将自己实现的学习算法的代码整理到其中
1. 数据预处理
   数据读取
   去噪（有大量 outliers，必须进行）：将离群值设置为缺失，交由第四步补全数据
   处理缺失数据：将缺失特征填补为数据该特征上的平均值、KNN 方法、过滤法（分别采用）
   降维：Relieff 方法、嵌入式选择
   标准化：通过缩放，将每个特征的均值调整为 0，方差调整为 1
2. 对每个模型获取训练算法的实现（整理到 funs.py 中）或调包
   线性回归模型（Lab1）：由于数据集较大出现指数下溢，因此调用其他的实现
   决策树模型（Lab3，调包）
   神经网络模型（调包）
   支持向量机（Lab2）
   XGBoost 模型（Lab3，调包）
3. 对每个模型都走一遍**调参**流程，选取最优的结果
   划分训练集和验证集
   设定参数进行训练
   根据训练好的模型进行验证，获得验证指标（采用**准确率**以及 **weighted-accuracy**）
   调整参数，重复测试（采用交叉验证的方法）
   保存模型
5. 根据最好的模型做预测，保存结果

四、实验结果

数据预处理部分：

训练集标签分布：

类别	0	1	2	3
数量	2523	2521	2500	2456

不难看出训练集的数据共含 4 种标签，且分布均匀

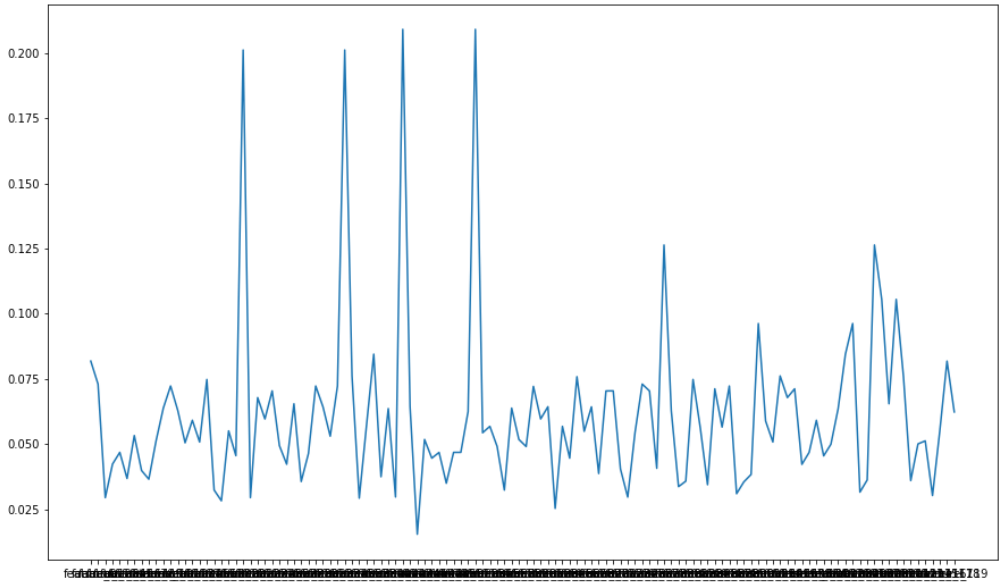
训练集特征分布：

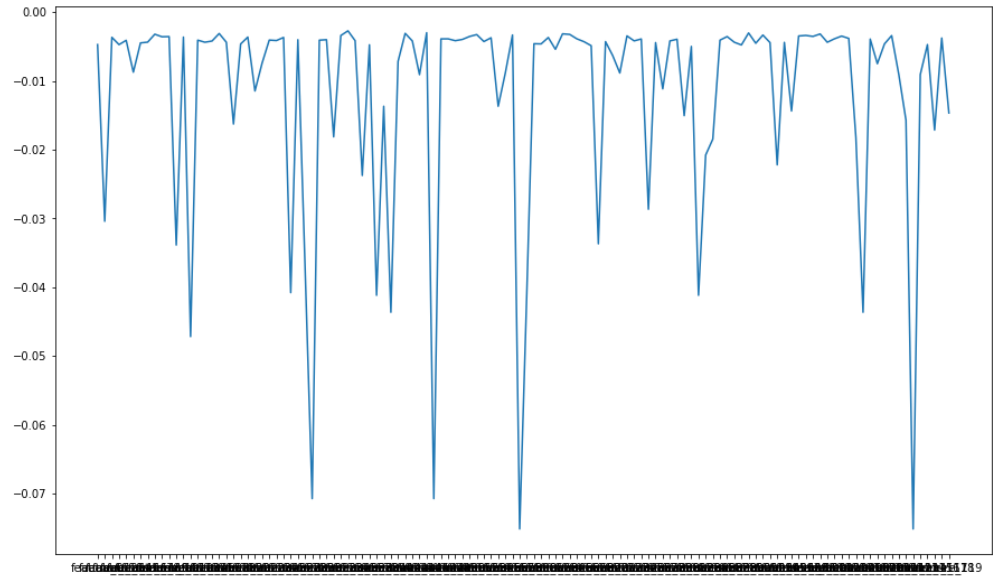
图略，详见附件 main.ipynb，各个特征单独绘制

大部分特征取值接近正态分布和均匀分布；少部分特征取值较少，可能是离散型特征。所有特征都有少量的缺失值以及离群点，且往往离群点与正常点相差几个数量级。若将所有带有缺失值和离群点的数据删去，只剩下 2415 个样本，这会浪费大量数据，对学习过程不利。

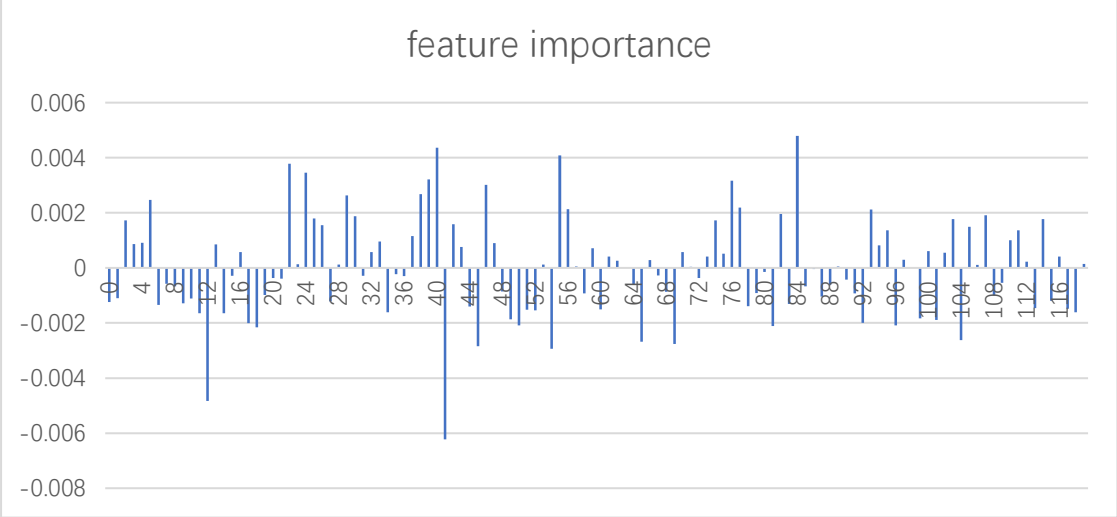
特征相关性：

下图分别是是各个特征与其他特征之间**相关系数**的最大值和最小值(绘图形式选取不是恰当，但仍能看出问题)，可以看出特征之间没有明显的相关性





特征选择：采用 Relieff 方法对特征进行筛选，得到特征权重如下：
（平均值法填充）



可以看出部分特征具有较大的权重，将它们筛选出来进行学习，可以有效提高训练速度
根据观察，选取前二十个特征进行学习

由于本次实验正确率非常不理想（采用了所提到的所有预处理方式，特征选取采用过滤式，嵌入式等方法，但泛化准确率总是在 0.24 到 0.27 之间浮动且不稳定，只有少数情况有所进展但是后来未能复现，如图所示），因此调参的标准只选准确率，且具体展开在此省略，只记录最优参数，详细过程在 main.ipynb 文件中。

```
X_train = r.transform(X)
result = cross_val_score(model_MLP, X_train, y, cv=5)
print("number of features: ", n_feature, '\t', "平均准确率: ", np.mean(result), '\t',
    "准确率标准差: ", np.std(result))
```

[28] ✓ 14m 43.1s

```
number of features:  5   平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  10          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  15          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  20          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  25          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  30          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  35          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  40          平均准确率:  0.5023      准确率标准差:  0.00024494897427829084
number of features:  45          平均准确率:  0.47969999999999996         准确率标准差:  0.02733422762764662
number of features:  50          平均准确率:  0.4729      准确率标准差:  0.024886542548132298
number of features:  55          平均准确率:  0.4690000000000003          准确率标准差:  0.027820855486487082
number of features:  60          平均准确率:  0.44799999999999995         准确率标准差:  0.027247018185482245
number of features:  65          平均准确率:  0.42889999999999995         准确率标准差:  0.007088018058667747
number of features:  70          平均准确率:  0.4285      准确率标准差:  0.012727922061357868
number of features:  75          平均准确率:  0.42469999999999997         准确率标准差:  0.006903622237637288
number of features:  80          平均准确率:  0.42749999999999994         准确率标准差:  0.014652644812456227
number of features:  85          平均准确率:  0.41879999999999995         准确率标准差:  0.006554387843269582
number of features:  90          平均准确率:  0.4107      准确率标准差:  0.013905394636614946
number of features:  95          平均准确率:  0.40390000000000004         准确率标准差:  0.010947145746723192
number of features:  100         平均准确率:  0.40920000000000006         准确率标准差:  0.008121576201698767
number of features:  105         平均准确率:  0.4067      准确率标准差:  0.00713862731902988
number of features:  110         平均准确率:  0.40490000000000004         准确率标准差:  0.0072346388990743495
number of features:  115         平均准确率:  0.4015000000000001          准确率标准差:  0.008712060605849793
number of features:  120         平均准确率:  0.396       准确率标准差:  0.014053469322555188
```

线性回归模型：
Penalty：L1　tol：10^{-1}　C：1　时得到最优准确率 0.2535

神经网络模型：（选择特征 20 个）
单隐藏层：学习率：10^{-5}，tol：10^{-3}，α：10^{-1} 时得到最优准确率 0.261
双隐藏层：学习率：10^{-2}，tol：10^{-6}，α：10^{-5} 时得到最优准确率 0.264

决策树：
划分判据：entropy
max_depth：7
ccp_alpha：1
最优准确率：0.2523

支持向量机：（甚至无法收敛，只能手动指定最大迭代次数）
kernel：线性核（与之相对的是 rbf）
Tol：0.001
C：0.5
最优准确率：0.2524

XGBoost：（15 个特征）
learning_rate：1
max_depth：4
colsample_bytree：1
subsample：0.8
最优准确率：0.2675

五、实验分析

本次实验准确率都不理想，主要有几个方面的问题：

1. 多分类问题中，准确率本身就是较为严苛的指标
2. 数据预处理时，缺失值和 outliers 可能含有重要信息被忽略
3. knn 方法和 relieff 算法本身对冗余特征效果不理想

对于实验结果，主要的改进方向还是数据预处理和调参，以及补上缺失的假设检验步骤