

# 机器学习

## 作业一

### 一. (30 points) 性能度量

学习器  $\mathcal{L}$  在某个多分类任务数据集上的预测混淆矩阵如表 1 所示，请回答下列问题。本题的答案请以分式或者小数点后两位的形式给出，比如  $P=0.67$ 。

真实情况	预测结果		
	第 1 类	第 2 类	第 3 类
第 1 类	7	1	4
第 2 类	2	6	4
第 3 类	2	2	8

表 1: 学习器  $\mathcal{L}$  在某个多分类任务数据集上的预测混淆矩阵

1. 该学习器的预测准确率是多少? (5 points)
2. 计算该学习器的微查准率 (micro-P)、宏查准率 (macro-P)、微查全率 (micro-R) 和宏查全率 (macro-R)。 (10 points)
3. 计算该学习器的微 F1 (micro-F1) 和宏 F1 (macro-F1)。 (5 points)
4. 在多分类中，每个样例只有一个标签。而在多标签分类中，每个样例可以有多个标签。如表 2 所示，一共有 3 个标签，样例  $x_1$  的标签是 1 和 3，学习器的预测是 1 和 2。请根据教材 2.3.2 查准率、查全率与 F1 章节的描述和表 2 的样例，计算学习器  $\mathcal{L}_1$  的微查准率、微查全率、微 F1、宏查准率、宏查全率和宏 F1。 [提示: 依然是利用各类的混淆矩阵计算微 F1 和宏 F1.] (10 points)

样例	$x_1$	$x_2$	$x_3$	$x_4$
标签	1,3	1,2	2,3	1,2,3
学习器预测	1,2	2,3	2,3	1,3

表 2: 学习器  $\mathcal{L}_1$  的样例表

## 二. (30 points) 性能度量

假设数据集包含 10 个样例, 其对应的真实标签和学习器的输出值 (从大到小排列) 如表 3 所示。该任务是一个二分类任务, 标签 1 或 0 表示真实标签为正例或负例。学习器的输出值代表学习器对该样例是正例的置信度 (认为该样例是正例的概率)。

样例	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
标签	1	1	0	1	1	0	1	0	0	0
学习器输出值	0.92	0.75	0.62	0.55	0.49	0.4	0.31	0.28	0.2	0.1

表 3: 样例表

1. 计算 P-R 曲线每一个端点的坐标并绘图 (8 points);
2. 计算 ROC 曲线每一个端点的坐标并绘图 (8 points);
3. 基于上一问, 计算 AUC 的值 (4 points); AUC 值请以小数点后两位的形式给出。
4. FPR95 是一个常见的性能度量指标, 它指的是当真正例率 (true positive rate) 为 95% 时, 假正例率 (false positive rate) 的数值。请问该指标越高学习性能越好还是越低性能越好, 并且求解 FPR75 为多少。[提示: FPR75 和 FPR95 类似, FPR75 是真正例率为 75%。](10 points)

## 三. (10 points) 规范化

Min-max 规范化和 z-score 规范化是两种常见的规范化方法。两种规范化方法分别如下面的公式所示:

$$x' = x'_{min} + \frac{x - x_{min}}{x_{max} - x_{min}} \times (x'_{max} - x'_{min}) \quad (1)$$

$$x' = \frac{x - \bar{x}}{\sigma_x} \quad (2)$$

其中  $x$  和  $x'$  分别是规范化前后的值,  $x_{max}$  和  $x_{min}$  分别是规范化前的最大值和最小值,  $x'_{max}$  和  $x'_{min}$  分别是规范化后的最大值和最小值,  $\bar{x}$  和  $\sigma_x$  是规范化前的均值和标准差。请分析二种规范化的优缺点。

## 四. (30 points) 线性回归

给定包含  $m$  个样例的数据集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ , 其中  $\mathbf{x}_i = (x_{i1}; x_{i2}; \dots; x_{id}) \in \mathbb{R}^d$ ,  $y_i \in \mathbb{R}$  为  $\mathbf{x}_i$  的实数标记。线性回归模型要求该线性模型的预测结果和其对应的标记之间的误差之和最小:

$$(\mathbf{w}^*, b^*) = \underset{(\mathbf{w}, b)}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^m (y_i - (\mathbf{w}^\top \mathbf{x}_i + b))^2 \quad (3)$$

即寻找一组权重  $(\mathbf{w}, b)$ , 使其对  $D$  中示例预测的整体误差最小。定义  $y = [y_1; y_2; \dots; y_m] \in \mathbb{R}^m$  且  $\mathbf{X} = [\mathbf{x}_1^\top; \mathbf{x}_2^\top; \dots; \mathbf{x}_m^\top] \in \mathbb{R}^{m \times d}$ .

1. 请将线性回归的优化过程使用矩阵进行表示; (10 points)
2. 使用矩阵形式给出权重  $\mathbf{w}^*$  和偏移  $b^*$  最优解的表达式; (10 points)
3. 针对波士顿房价预测数据集 (boston), 编程实现原始线性回归模型。请基于闭式解在训练集上构建模型, 计算测试集上的均方误差 (Mean Square Error, MSE)。请参考如下形式完成函数 `linear_regression` 和 `MSE` 的代码。除示例代码中使用到的 `sklearn` 库函数外, 不能使用其他的 `sklearn` 函数, 需要基于 `numpy` 实现线性回归模型闭式解以及 `MSE` 的计算. (10 points)

```
In [3]: from sklearn.datasets import load_boston
        from sklearn.model_selection import train_test_split

X, y = load_boston(return_X_y=True)
trainx, testx, trainy, testy = train_test_split(X, y, test_size = 0.33, random_state = 42)

def linear_regression(X_train, y_train):
    ''' 线性回归
        : 参数X_train: np.ndarray, 形状为(n, d), n 个d 维训练样本
        : 参数y_train: np.ndarray, 形状为(n, 1), 每个样本的标签
        : 返回: 权重矩阵w
    '''

def MSE(X_train, y_train, X_test, y_test):
    ''' 调用linear_regression得到权重矩阵w后计算MSE
        : 参数X_train: np.ndarray, 形状为(n, d), n 个d 维训练样本
        : 参数y_train: np.ndarray, 形状为(n, 1), 每个训练样本的标签
        : 参数X_test: np.ndarray, 形状为(m, d), m 个d 维测试样本
        : 参数y_test: np.ndarray, 形状为(m, 1), 每个测试样本的标签
        : 返回: 标量, MSE 值
    '''

linear_regression_MSE = MSE(trainx, trainy, testx, testy)
```