

# 非均衡分类问题

从准确度悖论谈起

小胖



# 目录

## ONE 非均衡数据集

准确度悖论

## TWO 解决办法

调整类别权重

## THREE 代码实现

scikit-learn

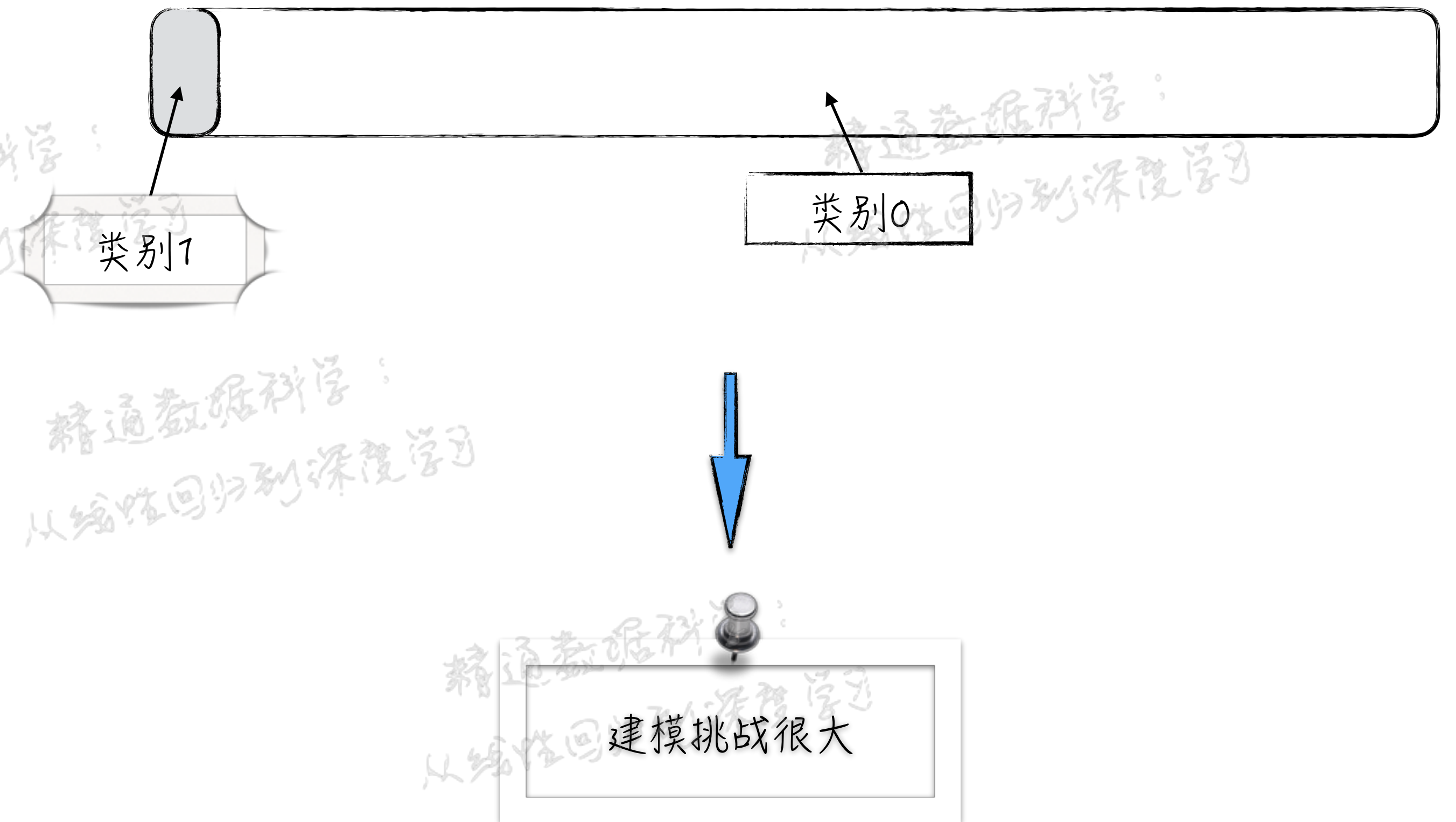
# 非均衡数据集

## 指标定义

各类别占比差别很大的数据集就是**非均衡数据集**

- 在实际的应用中，会经常遇到非均衡数据集，比如信贷、反欺诈、广告预测等
- 对于多元分类问题，如果使用OvR策略，也容易引发“潜在的”非均衡数据集

非均衡数据集会给**模型搭建带来困难**



# 非均衡数据集

## 准确度定义

准确度 = 预测准确的样本数 / 全体样本数

这个看似很直观的评估指标在面对非均衡分类问题时会严重失真

		真实值	
		1	0
预测值	1	真阳性 (true positive) TP	伪阳性 (false positive) FP
	0	伪阴性 (false negative) FN	真阴性 (true negative) TN

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

# 非均衡数据集

准确度悖论

非均衡分类问题

类别1：  
10个数据点

类别0：990个数据点

准确度悖论：  
模型A比模型B更好？

模型A		真实值	
		1	0
预测值	1	0	0
	0	10	990

$$ACC(A) = \frac{990+0}{990+10+0+0} = 99\%$$

模型B		真实值	
		1	0
预测值	1	9	90
	0	1	900

$$ACC(B) = \frac{900+9}{900+90+9+1} = 90.9\%$$



# 非均衡数据集

对建模的影响

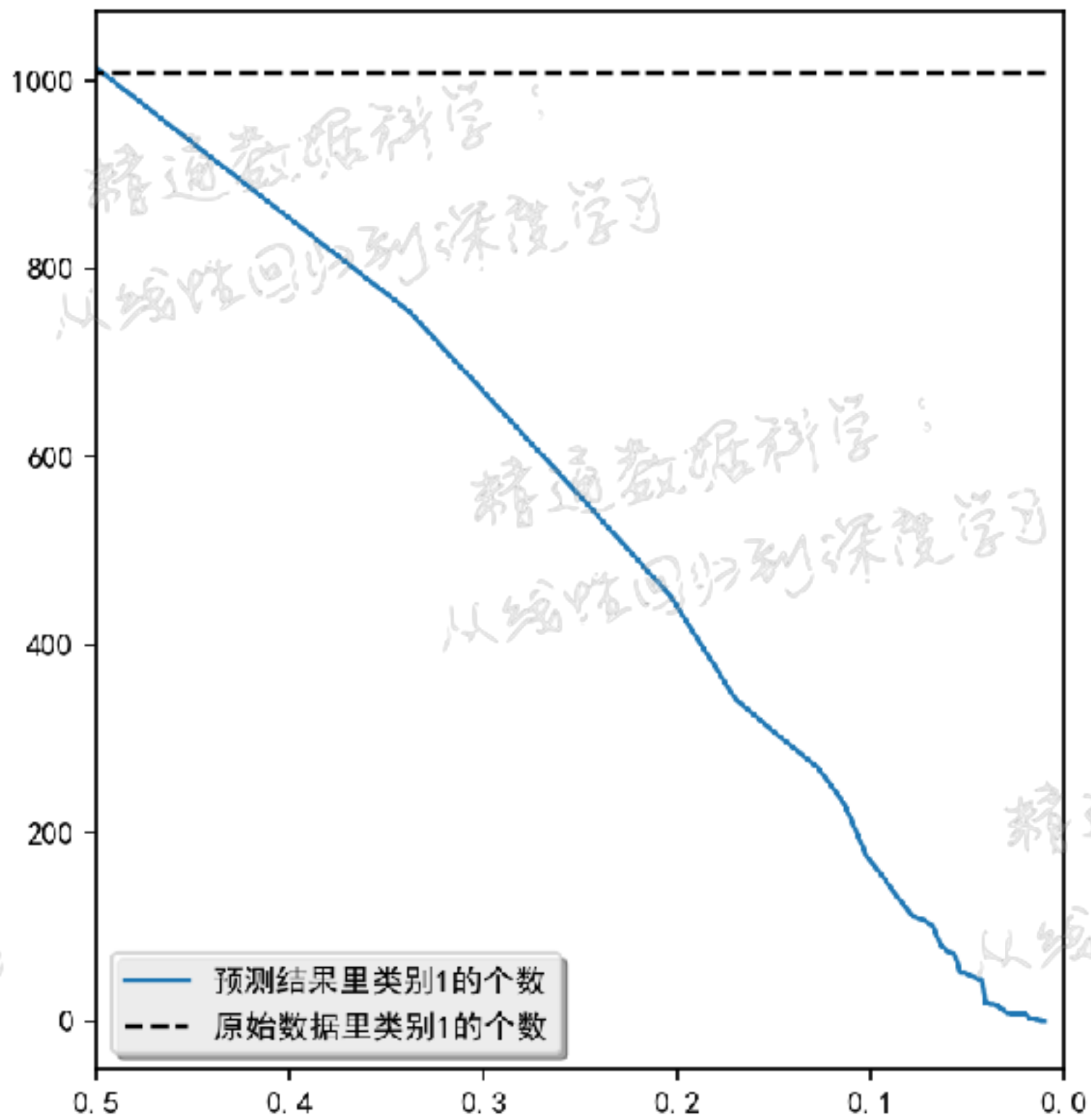
$$y = \begin{cases} 1, & x_1 - x_2 + \varepsilon > 0 \\ 0, & \text{else} \end{cases}$$

$\varepsilon$  是随机扰动项, 服从逻辑分布

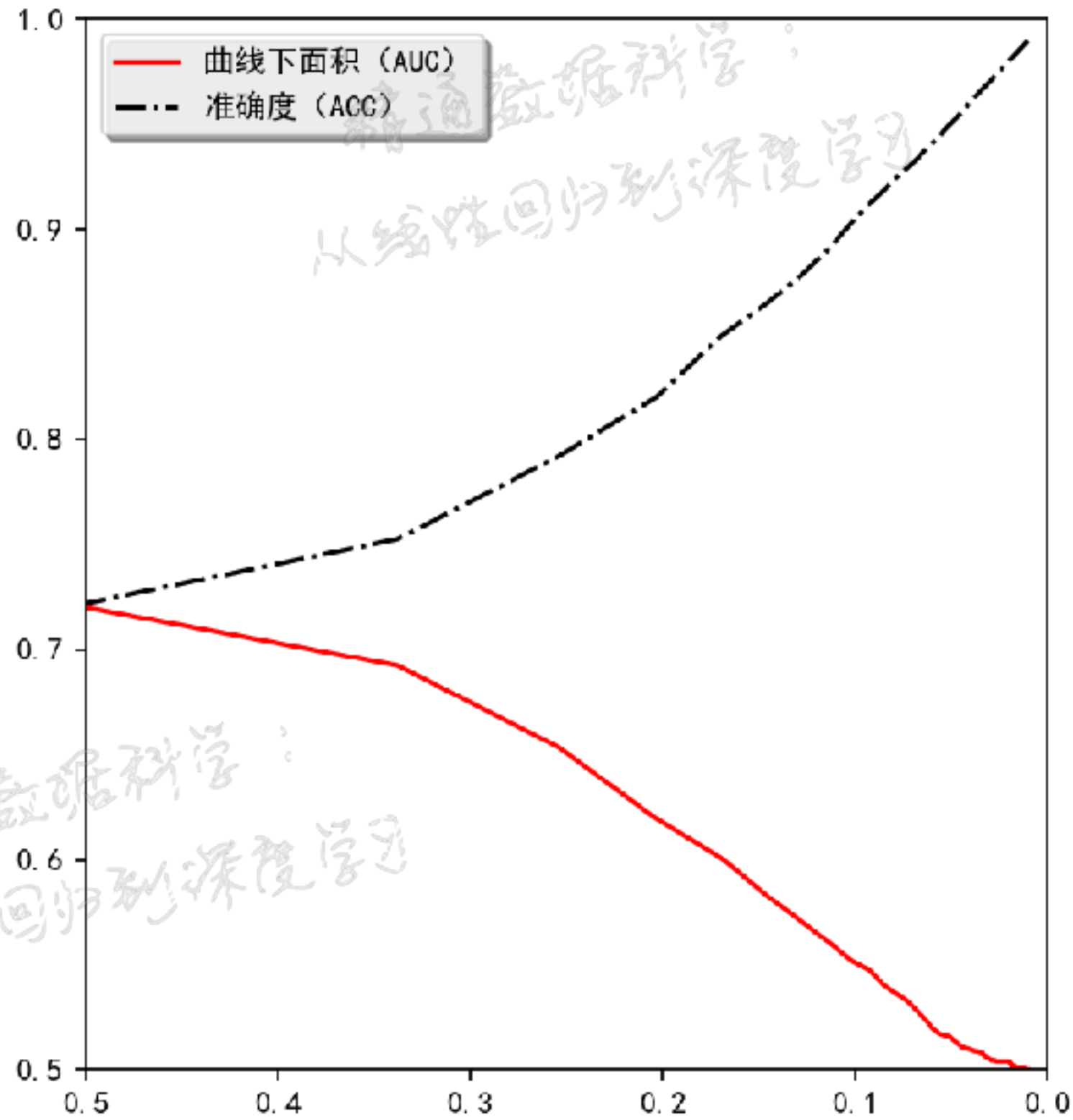
$y, x_1, x_2$  完美服从逻辑回归模型的假设

使用逻辑回归模型对数据进行建模

- 虽然y, x1, x2之间完美服从逻辑回归模型的假设, 但数据越不均衡, 模型效果越差
- 当面对非均衡数据集时, 准确度这个评估指标会严重失真



类别1所占比例



类别1所占比例

# 目录

## ONE 非均衡数据集

准确理论

## TWO 解决办法

调整类别权重

## THREE 代码实现

scikit-learn

# 解决办法

数学原因

逻辑回归的参数估计公式：

$$h(X_i) = 1 / (1 + e^{-X_i \beta})$$

$$\hat{\beta} = \operatorname{argmin}_{\beta} \sum_i -y_i \ln h(X_i) - (1 - y_i) \ln[1 - h(X_i)]$$

$h(X_i)$  靠近 1, 预测结果为类别 1;  
反之预测结果为 0

非均衡数据集：  
类别 1 少，类别 0 多

数据分布：

$h(X)$  靠近 0

每个点的权重都等于 1

$$-\ln[1 - h(X)]$$

$h(X)$  靠近 1

$$-\ln h(X)$$

● 类别 0  
○ 类别 1

$$X_1 \approx X_2 \approx \dots \approx X_6 \approx X$$

$$h(X) \approx 0$$

“牺牲”类别 1, “迁就”类别 0;  
模型预测结果几乎都为类别 0

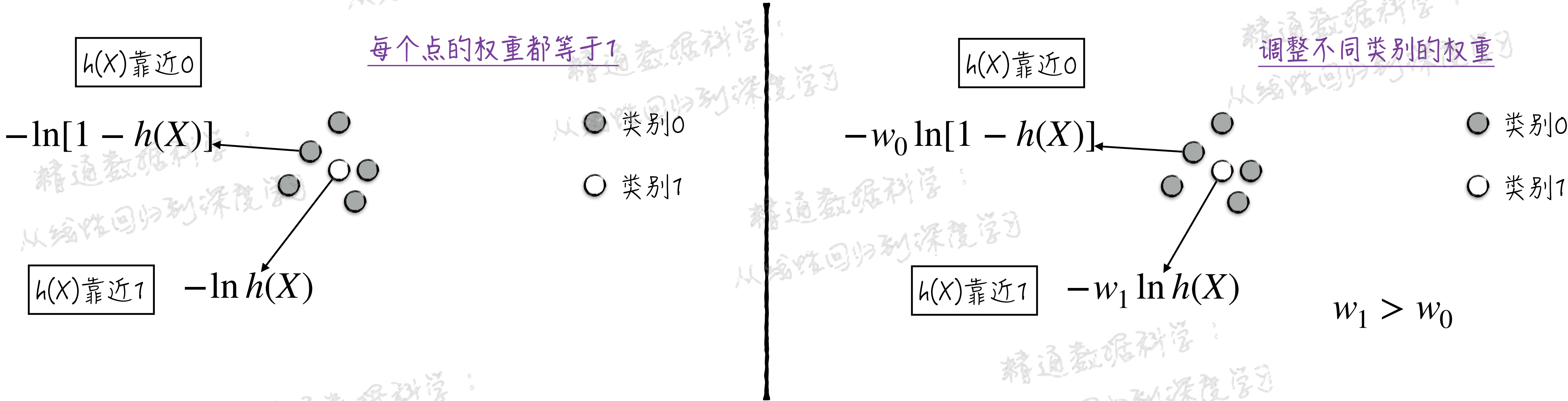
模型训练结果：



# 解决办法

调整类别权重

$$\hat{\beta} = \operatorname{argmin}_{\beta} \sum_i -y_i \ln h(\mathbf{X}_i) - (1 - y_i) \ln[1 - h(\mathbf{X}_i)]$$



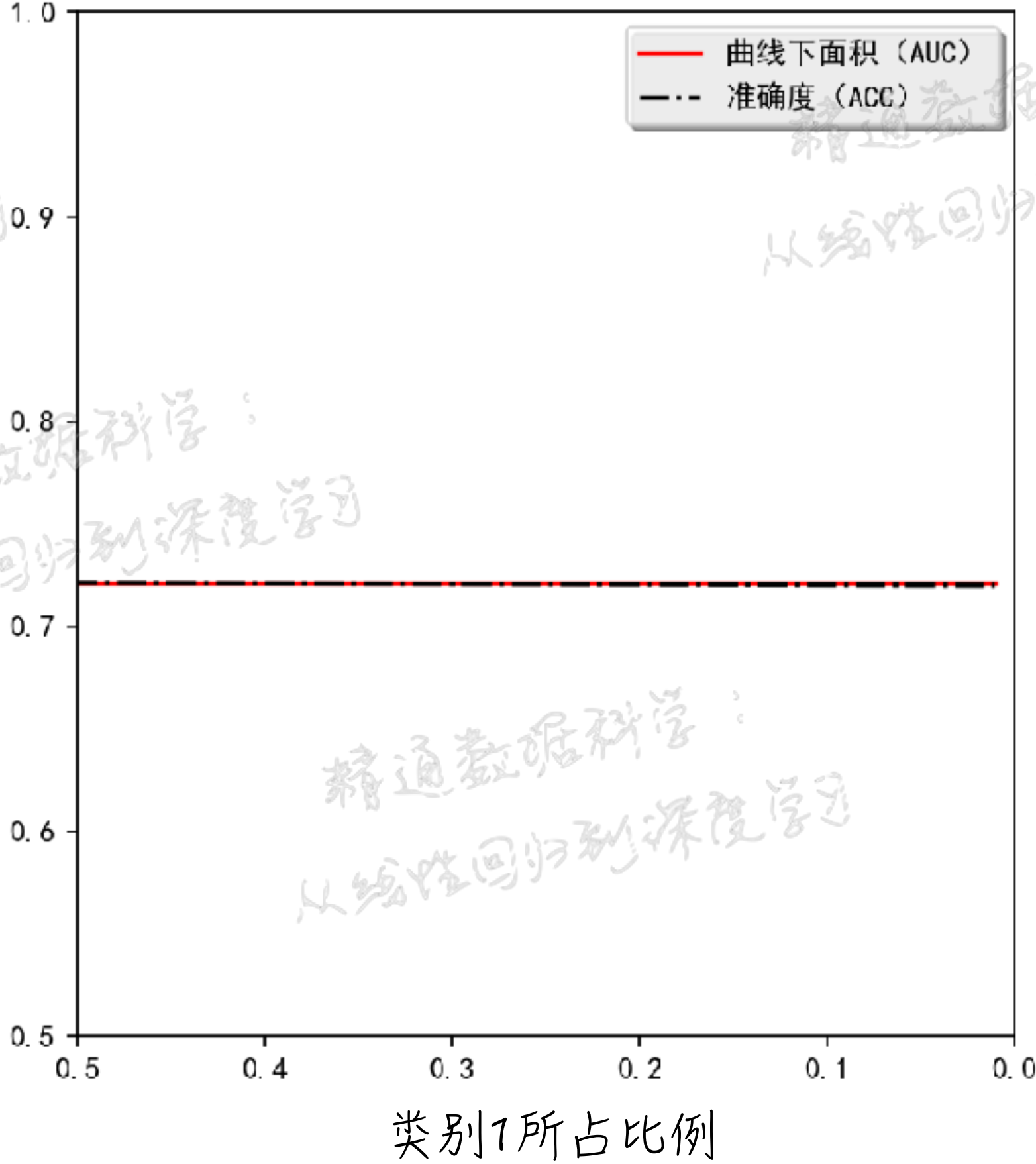
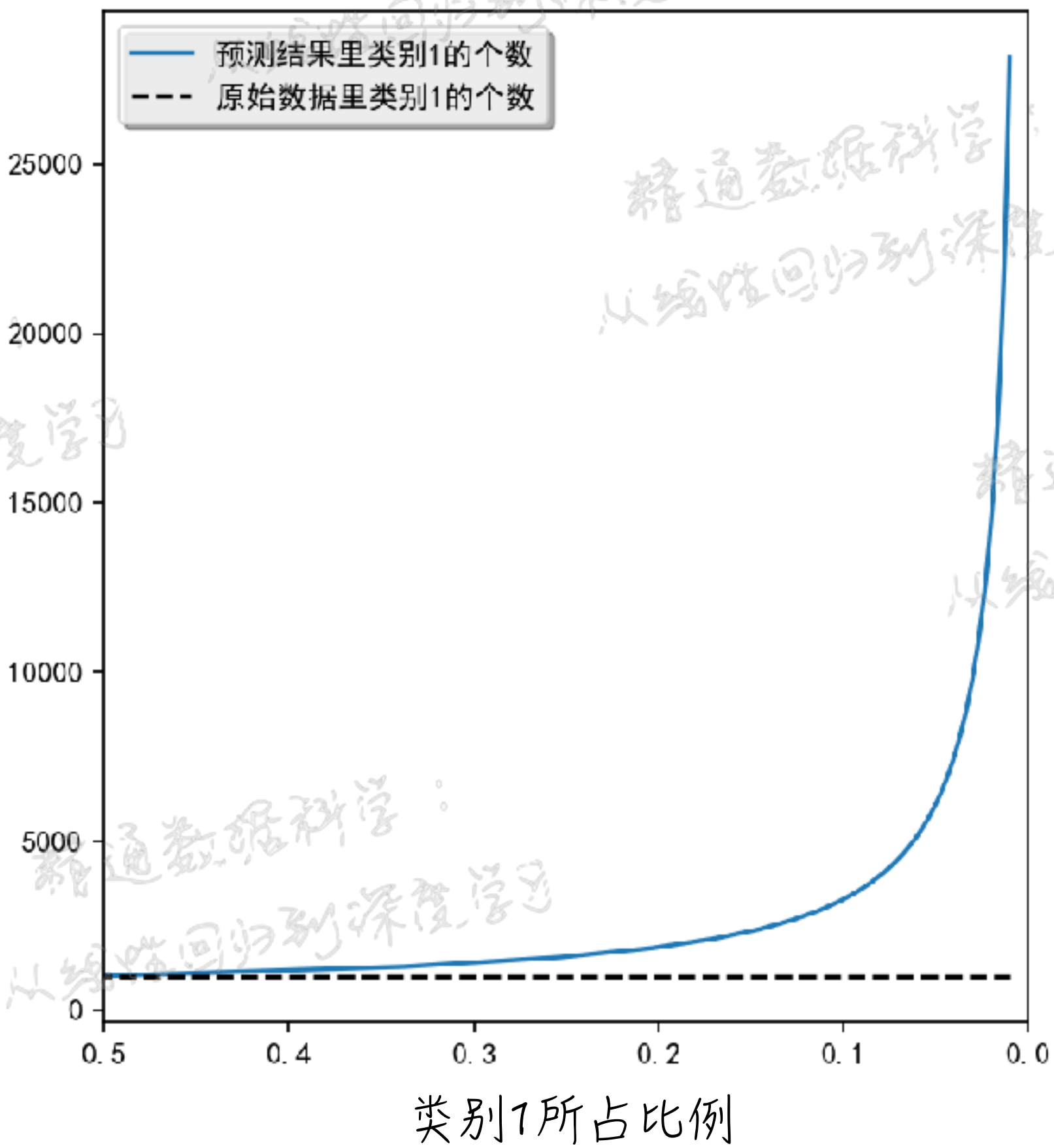
修正逻辑回归的  
损失函数

$$\hat{\beta} = \operatorname{argmin}_{\beta} \sum_i -w_1 y_i \ln h(\mathbf{X}_i) - w_0 (1 - y_i) \ln[1 - h(\mathbf{X}_i)]$$

# 解决办法

调整类别权重

平衡类别权重后的模型效果





# 目录

## ONE 非均衡数据集

准确理论

## TWO 解决办法

调整类别权重

## THREE 代码实现

scikit-learn

精通数据科学；  
从线性回归到深度学习

# THANK YOU