



中国科学技术大学
University of Science and Technology of China

《人工智能数学原理与算法》

逻辑回归实验

王翔


xiangwang@ustc.edu.cn

逻辑回归：贷款违约检测

实验目的

数据预处理: 理解pandas 进行数据清洗的基本方法，掌握 torch.utils.data.Dataset 和DataLoader 的实现细节及其在深度学习数据中的应用。

搭建逻辑回归模型，实现参数正则化防止过拟合。




01 分类问题

02 数据处理

03 回归算法



目录



01 分类问题

02 数据处理

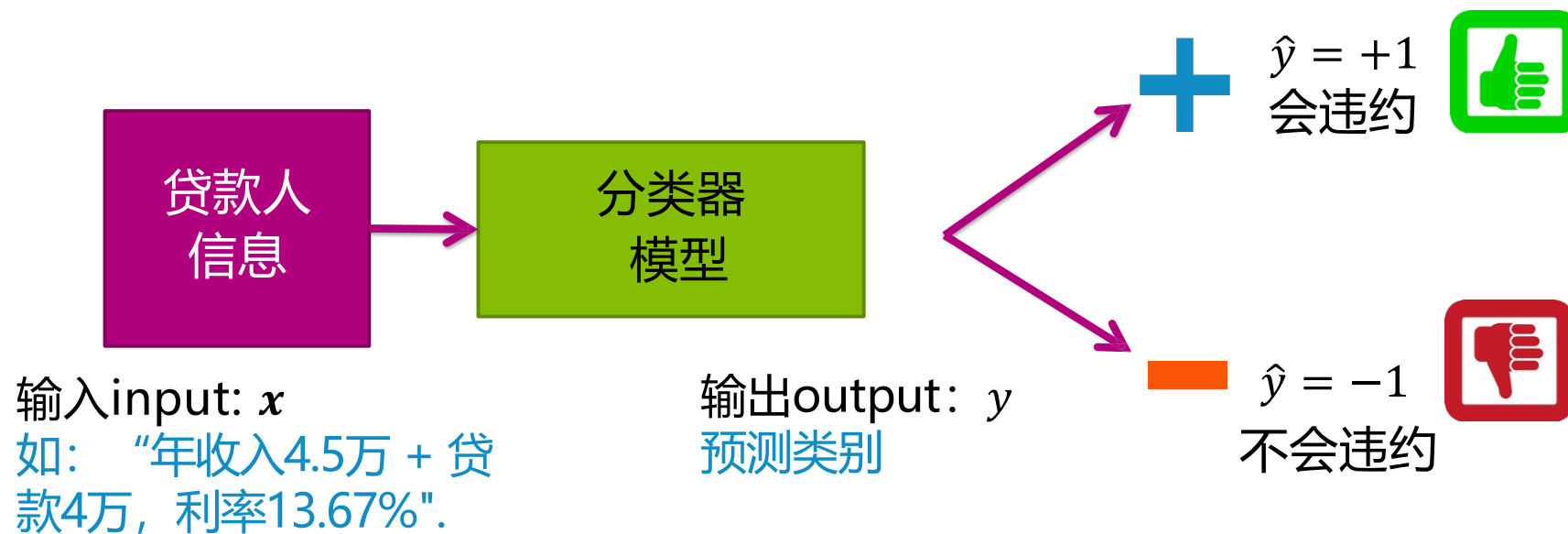
03 回归算法



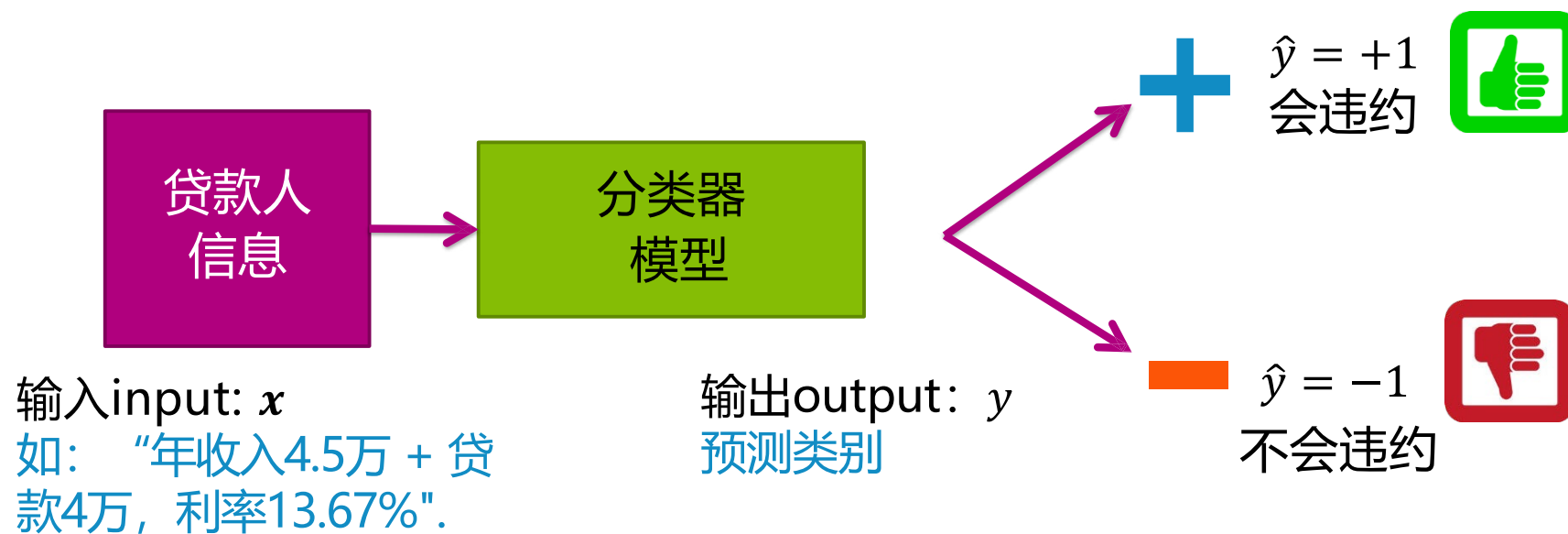
目录

逻辑回归：贷款违约检测

任务描述:数据来自某信贷平台的贷款记录，以个人信贷为背景，需要根据贷款申请人的数据信息预测其是否有违约的可能，以此判断是否通过此项贷款。



逻辑回归：贷款违约检测





01 分类问题

02 数据处理

03 回归算法

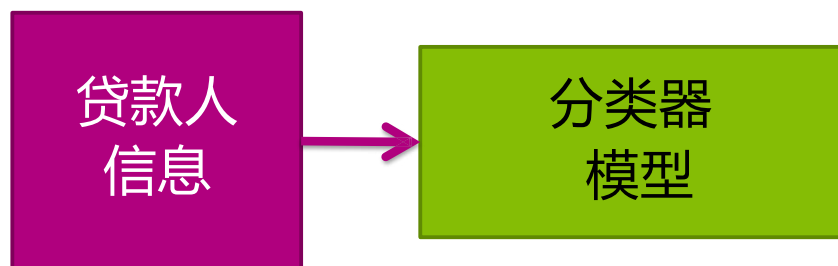


目录

逻辑回归：贷款违约检测

数据处理: 为了能够完成各种数据操作，我们需要pandas来存储和操作数据

数据介绍:



输入input: x

如: “年收入4.5万 + 贷款4万, 利率13.67%”.

输出output: y

预测违约

输入: 申请人的信息, 共47种特征, 包含了如贷款金额、贷款年限、贷款利率、年收入、工作年限等, 其中有15列匿名特征, 为申请人行为计数特征。

输出: 申请人贷款违约的可能性。

标签: “isDefault”特征, 共0, 1两种。1表示该贷款违约。

数据集描述: 共1万数据, 其中8000为训练集, 2000为测试集。训练集中50%是正样本, 即违约贷款。数值类变量42种, 33种连续型数值变量, 9种离散型数值变量。5种类别类特征, 有grade(A、B、C、D)、就业年限(<1 year, 1-3 year等), 还有时间类信息(yyyy-mm-dd)。

数据预处理:

逻辑回归：贷款违约检测

数据处理: 为了能够完成各种数据操作，我们需要pandas来存储和操作数据
数据预处理：

(1) 使用Pandas处理缺失值

(a) 插值法 :用替代值（如均值）填补缺失值。

(b) 删除法:直接忽略或删除含有缺失值的行或列。

(2) 使用torch dataset实现数据处理与加载

(a) `torch.utils.data.Dataset` :

(i) 重载`__len__()`返回数据集的大小.

(ii)重载`__getitem__()`返回数据集的特定项。

(b) `torch.utils.data.DataLoader`

(i) 封装Dataset类型作为迭代器



01 分类问题

02 数据处理

03 回归算法



目录

线性分类器：建模

$$\hat{y} = \text{sign}(\text{Score}(\mathbf{x}))$$

$$\begin{aligned}\text{Score}(\mathbf{x}) &= w_0 \cdot \phi(\mathbf{x})_0 + w_1 \cdot \phi(\mathbf{x})_1 + \dots + w_d \cdot \phi(\mathbf{x})_d \\ &= \sum_i w_i \cdot \phi(\mathbf{x})_i = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})\end{aligned}$$

特征提取器 $\boldsymbol{\phi}$:

特征1 = $\phi(\mathbf{x})_0$ (e.g., 1)

特征2 = $\phi(\mathbf{x})_1$ (e.g., $x[1]$ =年收入)

特征3 = $\phi(\mathbf{x})_2$ (e.g., $x[2]$ =贷款金额

or $\log(x[2])$

or $\log(x[2]/x[1])$)

特征d = $\phi(\mathbf{x})_d$ (其他关于 \mathbf{x} 的函数)

分类器：简单线性分类器

特征	系数
...	...



贷款人
信息



输入input: x

简单线性分类器：

$\text{Score}(x) = \text{各特征的加权和}$

If $\text{Score}(x) > 0$:

$$\hat{y} = +1$$

Else:

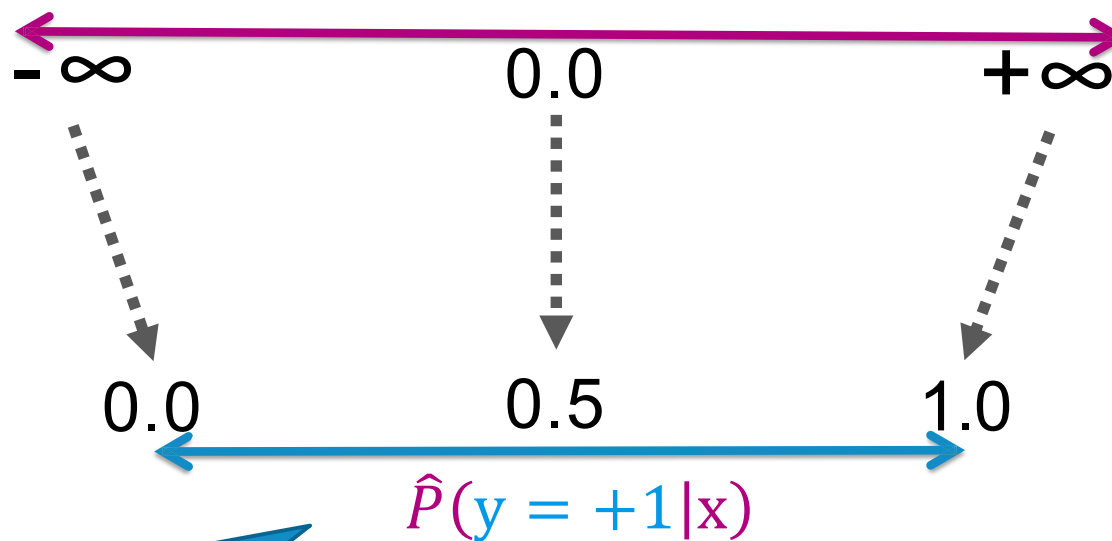
$$\hat{y} = 0$$

为什么不直接使用回归来构建分类器？

$$-\infty < \text{Score}(\mathbf{x}) < +\infty$$

$$\text{Score}(\mathbf{x}) = w_0 \cdot \phi(\mathbf{x})_0 + \dots + w_d \cdot \phi(\mathbf{x})_d = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})$$

我们如何将
 $-\infty, +\infty$
映射到 0, 1?



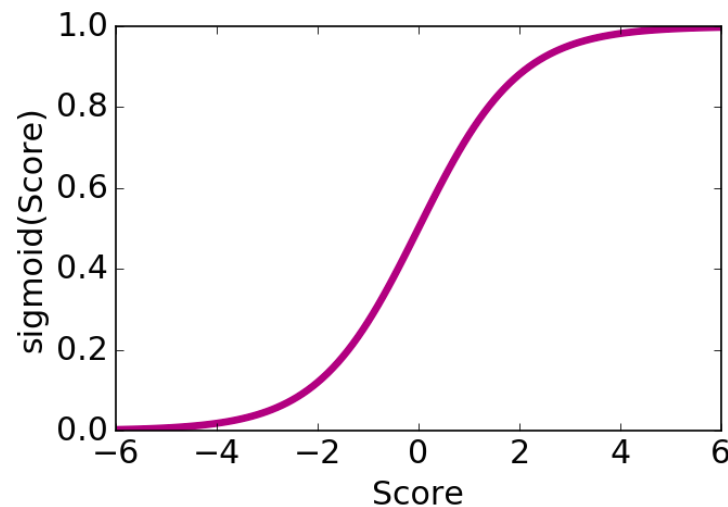
但概率处于 0 到 1 之间

逻辑回归：逻辑函数

□ 逻辑函数 (Logistic, 也称sigmoid, logit)

$$\text{sigmoid}(\text{Score}) = \frac{1}{1 + e^{-\text{Score}}}$$

Score	$-\infty$	-2	0.0	+2	$+\infty$
Sigmoid (Score)	0	0.12	0.5	0.88	1



逻辑回归 (logic Regression)

模型向量表示: $f_w(x) = \text{sigmoid}(\mathbf{w} \cdot \phi(x))$ $\mathbf{w} = [w_1, w_2]$ $\phi(x) = [1, x]$

参数向量/模型参数 特征提取器 特征向量

假设类: $\mathcal{F} = \{f_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^2\}$ (预测器 f 的集合)

损失函数: $\text{Loss}(f_w(x), y) = -[y * \log(f_w(x)) + (1 - y) * \log(1 - f_w(x))]$ 二项交叉熵损失 (Binary Cross Entropy loss)

$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \text{Loss}(x, y, \mathbf{w})$$

$$\arg \max_{\mathbf{w}} \ln p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \arg \min_{\mathbf{w}} \sum_{(x,y) \in \mathcal{D}} (y - \phi(x)\mathbf{w})^2$$

高斯假设下的最大似然估计 = 最小化二项交叉熵损失

逻辑回归中的过拟合

□ 使用L1惩罚进行稀疏逻辑回归

选择 \hat{w} 以最小化：

$$\ell(w) + \lambda \|w\|_1$$

调整参数 λ = 在拟合与参数规模之间平衡

L1正则的逻辑回归

使用以下方式选择 λ ：

- 验证集（适用于大型数据集）
- 交叉验证（适用于较小的数据集）
（如岭/套索回归）

逻辑回归中的过拟合

□ 使用L1惩罚进行稀疏逻辑回归

选择 $\hat{\mathbf{w}}$ 以最小化：

$$\ell(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

调整参数 λ = 在拟合与参数规模之间平衡

L2正则的逻辑回归

使用以下方式选择 λ ：

- 验证集（适用于大型数据集）
- 交叉验证（适用于较小的数据集）
（如岭/套索回归）