逻辑回归

黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

目录

Contents

- 逻辑回归简介
  - 应用场景，数学知识
- 逻辑回归原理
- 逻辑回归API函数和案例
- 分类问题评估
  - 混淆矩阵、精确率、召回率、F1-score、AUC指标、ROC曲线
- 电信客户流失预测案例

学习目标

Learning Objectives

1. 知道逻辑回归的应用场景
2. 复习逻辑回归应用到的数学知识

逻辑回归的应用场景

预测疾病 （是阳性、不是阳性）

银行信任贷款（放贷、还是不放贷）

情感分析 （正面、负面）

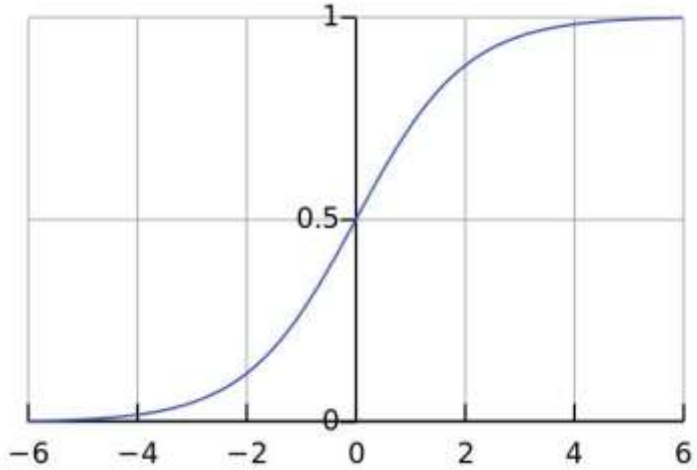
预测广告点击率 （点击、不点击）

逻辑回归是解决二分类问题的利器，你还能想到哪些二分类的场景？

逻辑回归数学基础 - sigmoid函数

- **sigmoid函数**
  - 数学公式 $$f(x) = \frac{1}{1 + e^{-x}}$$
  - 作用 把(-∞, +∞) 映射到 (0， 1)
  - 数学性质 单调递增函数

    拐点在x=0,y=0.5的位置
  - 导函数公式 $f'(x) = f(x)(1 - f(x))$



逻辑回归数学基础 - 概率

- **概率** - 事件发生的可能性

  联合概率和条件概率是概率论中的基本概念，它们用于描述随机变量之间的关系

  北京早上堵车的可能性 P_A = 0.7　　中午堵车的可能性 P_B = 0.3　　晚上堵车的可能性 P_C = 0.4

- **联合概率** - 指两个或多个随机变量同时发生的概率

  P_A = 0.7 周1早上 周2早上同时堵车的概率P_A P_B = 0.7 * 0.7 = 0.49

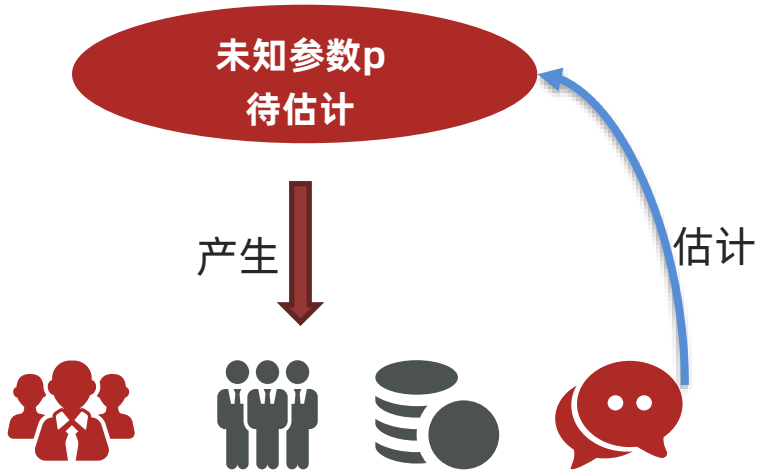
- **条件概率** -表示事件A在另外一个事件B已经发生条件下的发生概率，P(a|b)

  P_A = 0.7 周1早上 堵车的情况下，中午再堵车的概率 $P_{B|A}$ = 0.7 * 0.3 = 0.21

逻辑回归数学基础 - 极大似然估计

- 极大似然估计

核心思想：根据观测到的结果来估计模型算法中的未知参数



- 举个栗子

假设有一枚不均匀的硬币，出现正面的概率和反面的概率是不同的。假定出现正面的概率为θ，

抛了6次得到如下现象 D = {正面，反面，反面，正面，正面，正面}。每次投掷事件都是相互独立的。

则根据产生的现象D，来估计参数θ是多少？

$P(D|\theta)$ = P {正面，反面，反面，正面，正面，正面}

= P(正面|θ) P(反面|θ) P(反面|θ) P(正面|θ) P(正面|θ) P(正面|θ)

= θ *(1-θ)*(1-θ)*θ*θ*θ = $\theta^4(1-\theta)^2$

问题转化为：求此函数的极大值时，估计θ为多少！

$f(\theta) = \theta^4(1-\theta)^2$ 令导数=0求极值，可估计出θ值

$\frac{\partial f(\theta)}{\partial \theta} = 4\theta^{4-1}(1-\theta)^2 + \theta^4 2(1-\theta)(-1)$

$= 4\theta^3(1-\theta)^2 - 2\theta^4(1-\theta) = \theta^3(1-\theta)(4-6\theta) = 0$

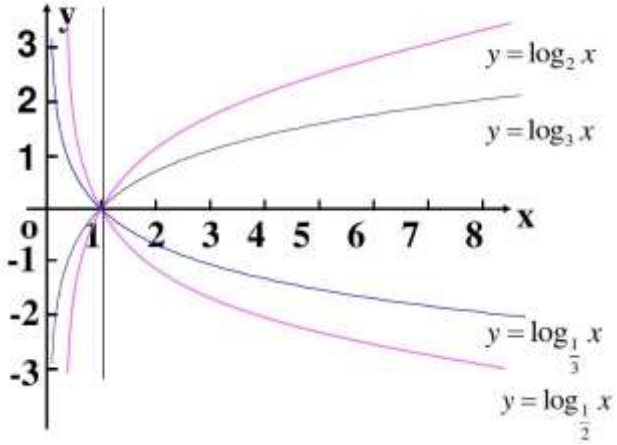
从而得：$\theta_1 = 0$ ， $\theta_2 = 1$， $\theta_3 = 2/3$； 取$\theta_3 = 2/3$

逻辑回归数学基础 - 对数函数

- 对数函数

  如果$a^b = N$ (a > 0, b ≠1), 那么b叫做以a为底N的对数。记为 $b = \log_a N$

  eg: $\log_{10} 100 = 2$　　$\log_2 16 = 4$　　注意：a > 1 a < 1时对数函数的图像



- 对数函数性质

$$①\log_a MN = \log_a M + \log_a N$$

$$②\log_a \frac{M}{N} = \log_a M - \log_a N$$

$$③\log_a M^n = n\log_a M \qquad \text{其中a>0,a≠0,M>0,N>0}$$

从对数运算性质来看：能把几个概率联乘的式子，改成log相加的形式

总结

1 逻辑回归的作用？

分类：二分类

2 激活函数 sigmoid 作用？

作用：把数值 映射到 (0， 1）

3 极大似然估计

通过极大化概率事件，来估计最优参数

4 对数函数

$$①\log_a MN = \log_a M + \log_a N$$

$$③\log_a M^n = n\log_a M$$

练习

设总体 X 的概率密度函数是 $f(x;\mu)=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x-\mu)^2}$，$-\infty<x<+\infty$

$x_1,x_2,\ldots,x_n$ 是一组样本值，求参数 μ 的最大似然估 计。

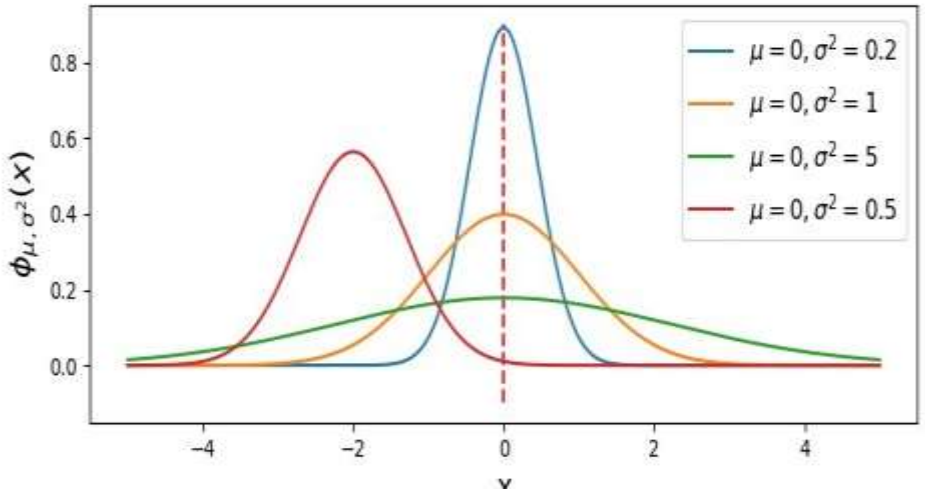
解：似然函数

$$L=\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x_i-\mu)^2}=\frac{1}{\left(\sqrt{2\pi}\right)^n}\exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^2\right\}$$

对似然函数两边，同时取ln $\ln L=-\frac{n}{2}\ln(2\pi)-\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^2$

对参数u求导函数 $\frac{d\ln L}{d\mu}=\sum_{i=1}^{n}(x_i-\mu)=0$ 当x =μ 时 概率最大

μ 为x的均值 $\hat{\mu}=\frac{1}{n}\sum_{i=1}^{n}x_i=\bar{x}$



目录

Contents

学习目标

Learning Objectives

1. 理解逻辑回归算法的原理
2. 知道逻辑回归的损失函数

逻辑回归原理 - 概念

- 逻辑回归概念 Logistic Regression
  - 一种分类模型，把线性回归的输出，作为逻辑回归的输入。
  - 输出是（0, 1）之间的值
- 基本思想
  1. 利用线性模型 $f(x) = w^Tx + b$ 根据特征的重要性计算出一个值
  2. 再使用 sigmoid 函数将 f(x) 的输出值映射为概率值

  （1）设置阈值(eg：0.5)，输出概率值大于 0.5，则将未知样本输出为 1 类

  （2）否则输出为 0 类
- 逻辑回归的假设函数

  $h(w) = sigmoid(w^Tx + b)$

  线性回归的输出，作为逻辑回归的输入

逻辑回归原理 - 概念

- 举个栗子：逻辑回归预测过程 （阈值为0.6）

样本特征值输入				回归		逻辑回归结果		预测结果	真实结果
12.3	20.0	16		82.4		0.4		B	A
9.4	21.1	7.2	回归计算	89.1	sigmoid	0.68		A	B
34.4	18.7	8.1	x W =	80.2	→	0.41	→	B	A
10.2	16.0	12.5		81.3		0.55		B	B
5.6	10.0	6.3		90.4		0.71		A	A

假设得出概率值是属于A的概率值

逻辑回归原理 - 损失函数

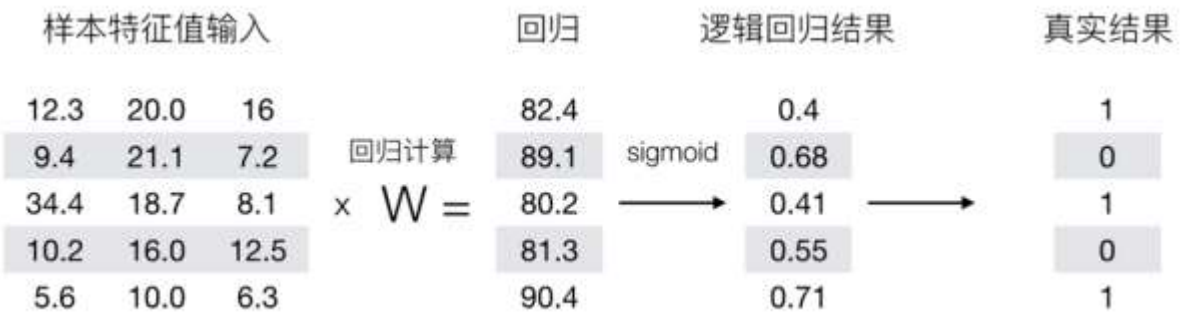
- 损失函数

$$Loss(L) = -\sum_{i=1}^{m}(y_i \log(p_i) + (1-y_i)\log(1-p_i))$$

p_i = sigmoid($w^T x + b$)是逻辑回归的输出结果

损失函数的工作原理：每个样本预测值有A、B两个类别，真实类别对应的位置，概率值越大越好

- 举个栗子 – 损失函数手工计算



计算损失：= [1log0.4 + (1-1)log(1-0.4) + # 第1个样本产生的损失

0log0.6 + (1-0)log(1-0.6) + # 第2个样本产生的损失

1log0.41 + (1-1)log(1-041) + ...

损失函数

- 1、1个样本的概率表示

  假设：有 0、1 两个类别，某个样本被分为 1 类的概率为 p，
  则分为 0 类的概率为 1-p，则每一个样本分类正确的概率为：

$$L = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{if } y = 0 \end{cases}$$

  样本类别为y=1概率是p、样本类别y=0概率是(1-p), 合成一个式子

$$L = p^y(1-p)^{1-y}$$

  我们对损失函数的希望是：当样本是1类别，模型预测的p越大越好；
  当样本是0类别，模型预测的(1-p)越大越好；

- 2、n个样本的概率表示

  假设：有样本[$(x_1, y_1), (x_1, y_2) \ldots, (x_n, y_n)$]，n个样本，所有样本都预测正确的概率为：

$$P = P(y_1|x_1)P(y_2|x_2)\ldots P(y_n|x_n) = \prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

  1.p_i 表示每个样本被分类正确时的概率

  2.y_i 表示每个样本的真实类别（0或1）

- 问题转化为：让联合概率事件最大时，估计w、b的权重参数，这就是极大似然估计

损失函数

- 3、极大似然函数转对数似然函数，取log优化函数：连乘形式转换为对数加法形式

$$H(L) = \sum_{i=1}^{m} \left(y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right)$$

$$p_i = \frac{1}{1 - e^{-(w^T x + b)}}$$

最大化问题将其变为最小化问题：

$$Loss(L) = - \sum_{i=1}^{m} \left(y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right)$$

- 4、使用梯度下降优化算法，更新逻辑回归算法的权重参数。

总结

1 逻辑回归原理

- 思想：解决分类问题，把线性回归的输出作为逻辑回归的输入

2 逻辑回归的损失函数 - 对数似然损失

- 数学表达式：Loss(L) = - $\sum_{i=1}^{m}$ （ $y_i \log(p_i)$ + $(1-y_i)\log(1-p_i)$)
- 损失函数设计思想：预测值为A、B 2个类别，真实类别所在的位置，概率值越大越好

练习

1 以下关于逻辑回归的说法正确的是？（**多选**）

A）逻辑回归应用在分类场景中

B）逻辑回归使用了回归将特征数据进行拟合

C）逻辑回归使用了sigmoid激活函数将回归的结果映射到了(0, 1)值域中

D）逻辑回归的损失函数使用了极大似然估计，因为其输出值是一个概率

答案：ABCD

目录

Contents

- 逻辑回归简介

  应用场景，数学知识

- 逻辑回归原理

- 逻辑回归API函数和案例

- 分类问题评估

  混淆矩阵、精确率、召回率、F1-score、AUC指标、ROC曲线

- 电信客户流失预测案例

学习目标

Learning Objectives

1. 知道逻辑回归的API
2. 动手实现癌症分类案例

逻辑回归API函数和案例 - API介绍

sklearn.linear_model.LogisticRegression(solver='liblinear', penalty='l2', C = 1.0)

- **solver 损失函数优化方法**:

  1 liblinear 对小数据集场景训练速度更快，sag 和 saga 对大数据集更快一些。

  2 正则化：

  （1）sag、saga 支持 L2 正则化或者没有正则化

  （2）liblinear 和 saga 支持 L1 正则化

- **penalty**：正则化的种类，L1 或者 L2
- **C**：正则化力度
- 默认将类别数量少的当做正例

逻辑回归API函数和案例 - 案例癌症分类预测

- **数据描述**

（1）699条样本，共11列数据，第一列用语检索的id，后9列分别是与肿瘤相关的医学特征，最后一列表示肿瘤类型的数值。

（2）包含16个缺失值，用”?”标出。

（3）2表示良性，4表示恶性

	A	B	C	D	E	F	G	H	I	J	K
1	Sample code number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
2	1000025	5	1	1	1	2	1	3	1	1	2
3	1002945	5	4	4	5	7	10	3	2	1	2
4	1015425	3	1	1	1	2	2	3	1	1	2
5	1016277	6	8	8	1	3	4	3	7	1	2
6	1017023	4	1	1	3	2	1	3	1	1	2
7	1017122	8	10	10	8	7	10	9	7	1	4
8	1018099	1	1	1	1	2	10	3	1	1	2
9	1018561	2	1	2	1	2	1	3	1	1	2
10	1033078	2	1	1	1	2	1	1	1	5	2
11	1033078	4	2	1	1	2	1	2	1	1	2
12	1035283	1	1	1	1	1	1	3	1	1	2
13	1036172	2	1	1	1	2	1	2	1	1	2
14	1041801	5	3	3	3	2	3	4	4	1	4
15	1043999	1	1	1	1	2	3	3	1	1	2

逻辑回归API函数和案例 - 癌症分类预测

- **案例分析**

1.导入依赖包

2.加载数据及数据预处理

2.1 缺失值处理

2.2 确定特征值,目标值

2.3 分割数据

3.特征工程(标准化)

4.模型训练，机器学习(逻辑回归)

5.模型预测和评估

逻辑回归API函数和案例 - 癌症分类预测

- **代码实现**

```
# 1.导入依赖包
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import numpy as np
```

```
def dm_LogisticRegression():
    # 2.加载数据及数据预处理
    data = pd.read_csv('./data/breast-cancer-wisconsin.csv')
    data.info()
    # 2.1 缺失值处理
    data = data.replace(to_replace="?", value=np.NAN)
    data = data.dropna()
    # 2.2 确定特征值,目标值
    x = data.iloc[:, 1:-1]
    print('x.head()-->\n', x.head())
    y = data["Class"]
    print('y.head()-->\n', y.head())
    # 2.3 分割数据
    x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=22)

    # 3.特征工程(标准化)
    transfer = StandardScaler()
    x_train = transfer.fit_transform(x_train)
    x_test = transfer.transform(x_test)

    # 4.模型训练，机器学习(逻辑回归)
    estimator = LogisticRegression()
    estimator.fit(x_train, y_train)

    # 5.模型预测和评估
    y_predict = estimator.predict(x_test)
    print('y_predict-->', y_predict)
    accuracy = estimator.score(x_test, y_test)
    print('accuracy-->', accuracy)
     print(estimator.score(x_test, y_predict))
```

练习

1、下列关于逻辑回归API的使用正确的是？（**多选**）

A）需要在sklearn的线性模型linear_model中导出使用

B）可以通过solver参数指定损失的优化方法

C）可以通过penalty参数指定使用哪种正则化方式

D）它默认将样本中类别数较多的一类当做正例

答案解析：它默认将样本中类别数较少的一类当做正例

答案：ABC

目录

Contents

- 逻辑回归简介

  应用场景，数学知识

- 逻辑回归原理

- 逻辑回归API函数和案例

- 分类问题评估

  混淆矩阵、精确率、召回率、F1-score、AUC指标、ROC曲线

- 电信客户流失预测案例

学习目标

Learning Objectives

1. 理解混淆矩阵的构建方法
2. 掌握精确率，召回率和F1score的计算方法

1. 只做预测准确率能满足各种场景需要吗？
2. 比如上述癌症检测的案例

   癌症患者有没有被全部预测（检测）出来。

分类评估方法 - 混淆矩阵

- 什么是混淆矩阵？

真实结果 \ 预测结果	正例	假例
正例	真正例TP	伪反例FN
假例	伪正例FP	真反例TN

- 混淆矩阵四个指标
  - 真实值是 正例 的样本中，被分类为 正例 的样本数量有多少，叫做真正例（TP，True Positive）
  - 真实值是 正例 的样本中，被分类为 假例 的样本数量有多少，叫做伪反例（FN，False Negative）
  - 真实值是 假例 的样本中，被分类为 正例 的样本数量有多少，叫做伪正例（FP，False Positive）
  - 真实值是 假例 的样本中，被分类为 假例 的样本数量有多少，叫做真反例（TN，True Negative）

分类评估方法 - 混淆矩阵

- 混淆矩阵-举个栗子

已知：样本集10样本，有 6 个恶性肿瘤样本，4 个良性肿瘤样本，我们假设恶性肿瘤为正例

模型A：预测对了 3 个恶性肿瘤样本，4 个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：3

2.伪反例 FN 为：3

3.伪正例 FP 为：0

4.真反例 TN：4

模型B：预测对了 6 个恶性肿瘤样本，1个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：6

2.伪反例 FN 为：0

3.伪正例 FP 为：3

4.真反例 TN：1

注意：TP+FN+FP+TN = 总样本数量

分类评估方法 - 混淆矩阵

- 混淆矩阵 - 举个栗子

```
# 1.导入依赖包
from sklearn.metrics import confusion_matrix
import pandas as pd

def dm01_混淆矩阵四个指标():
    # 2.构建数据，样本集中共有6个恶性肿瘤样本, 4个良性肿瘤样本
    y_true = ["恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "良性", "良性", "良性", "良性"]

    # 3.1 混淆矩阵，模型A: 预测对了3个恶性肿瘤样本, 4个良性肿瘤样本
    print("模型A:")
    print("-" * 13)
    y_pred1= ["恶性", "恶性", "恶性", "良性", "良性", "良性", "良性", "良性", "良性", "良性"]
    result = confusion_matrix(y_true, y_pred,1 labels=["恶性", "良性"])
    print(pd.DataFrame(result, columns=["恶性(正例)", "良性(反例)"],  index=["恶性(正例)", "良性(反例)"])

    # 3.2 混淆矩阵，模型B: 预测对了6个恶性肿瘤样本, 1个良性肿瘤样本
    print("模型B:")
    print("-" * 13)
    y_pred2= ["恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "良性"]

    result = confusion_matrix(y_true, y_pred2, labels=labels)
    print(pd.DataFrame(result,  columns=dataframe_labels，    index=dataframe_labels))
```

分类评估方法 - 精确率、召回率、F1-score

- 精确率(Precision)
  - 查准率，对正例样本的预测准确率。比如：把恶性肿瘤当做正例样本，想知道模型对恶性肿瘤的预测准确率。
  - 计算方法：$P = \frac{TP}{TP + FP}$

真实结果 \ 预测结果	正例	假例
正例	真正例TP	伪反例FN
假例	伪正例FP	真反例TN

- 精确率(Precision) - 举个栗子

已知：样本集10样本，有 6 个恶性肿瘤样本，4 个良性肿瘤样本，我们假设恶性肿瘤为正例

模型A：预测对了 3 个恶性肿瘤样本，4 个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：3

2.伪反例 FN 为：3

3.伪正例 FP 为：0

4.真反例 TN：4

精度：3 / (3+0) = 100%

模型B：预测对了 6 个恶性肿瘤样本，1个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：6

2.伪反例 FN 为：0

3.伪正例 FP 为：3

4.真反例 TN：1

精度：6/(6+3) = 67%

分类评估方法 - 精确率、召回率、 F1-score

- 精确率(Precision) - 举个栗子

```
# 1.导入依赖包
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score

def dm01_精度Precision():

    # 2.构建数据，样本集中共有6个恶性肿瘤样本, 4个良性肿瘤样本
    y_true = [“恶性”, “恶性”, “恶性”, “恶性”, “恶性”, “恶性”, “良性”, “良性”, “良性”, “良性”]

    # 3.1 模型精确率评估，模型A: 预测对了3个恶性肿瘤样本, 4个良性肿瘤样本
    y_pred1 = [“恶性”, “恶性”, “恶性”, “良性”, “良性”, “良性”, “良性”, “良性”, “良性”, “良性”]
    result = precision_score(y_true, y_pred1, pos_label=“恶性”)
    print(“模型A精度:”, result)

    # 3.2 模型精确率评估，模型B: 预测对了6个恶性肿瘤样本, 1个良性肿瘤样本
    y_pred2 = ["恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "良性"]
    result = precision_score(y_true, y_pred2, pos_label="恶性")
    print("模型B精度:", result)
```

分类评估方法 - 精确率、召回率、 F1-score

- 召回率(Recall) - 概念
  - 也叫查全率，指的是预测为真正例样本占所有真实正例样本的比重

    例如：恶性肿瘤当做正例样本，则我们想知道模型是否能把所有的恶性肿瘤患者都预测出来。
  - 计算方法：$P = \frac{TP}{TP+FN}$

		预测结果	
		正例	假例
真实结果	正例	真正例TP	伪反例FN
	假例	伪正例FP	真反例TN

- 召回率(Recall) - 举个栗子

已知：样本集10样本，有 6 个恶性肿瘤样本，4 个良性肿瘤样本，我们假设恶性肿瘤为正例

模型A：预测对了 3 个恶性肿瘤样本，4 个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：3

2.伪反例 FN 为：3

3.伪正例 FP 为：0

4.真反例 TN：4

精度：3 / (3+0) = 100%

召回率：3 / (3 + 3) = 50%

模型B：预测对了 6 个恶性肿瘤样本，1个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：6

2.伪反例 FN 为：0

3.伪正例 FP 为：3

4.真反例 TN：1

精度：6/(6+3) = 67%

召回率：6 / (6 + 0) = 100%

分类评估方法 - 精确率、召回率、 F1-score

- 召回率- 举个栗子

```
# 1.导入依赖包
from sklearn.metrics import recall_score

def dm02_召回率recall():
    # 2.构建数据，样本集中共有6个恶性肿瘤样本, 4个良性肿瘤样本
    y_true = [“恶性”, “恶性”, “恶性”, “恶性”, “恶性”, “恶性”, “良性”, “良性”, “良性”, “良性”]

    # 3.1 模型召回率评估，模型A: 预测对了3个恶性肿瘤样本, 4个良性肿瘤样本
    y_pred1 = [“恶性”, “恶性”, “恶性”, “良性”, “良性”, “良性”, “良性”, “良性”, “良性”, “良性”]
    result = recall_score(y_true, y_pred1, pos_label=“恶性”)
    print(“模型A召回率:”, result)

    # 3.2 模型召回率评估，模型B: 预测对了6个恶性肿瘤样本, 1个良性肿瘤样本
    y_pred2= ["恶性", "恶性", "恶性", "恶性", "恶性", "恶性","恶性", "恶性", "恶性", "良性"]
    result = recall_score(y_true, y_pred2, pos_label="恶性")
    print("模型B召回率:", result)
```

分类评估方法 - 精确率、召回率、F1-score

- F1-score
  - 若对模型的精度、召回率都有要求，希望知道模型在这两个评估方向的综合预测能力？
  - 计算方法：$P = \frac{2 * Precision * Recall}{Precision + Recall}$

- F1-score - 举个栗子

已知：样本集10样本，有 6 个恶性肿瘤样本，4 个良性肿瘤样本，我们假设恶性肿瘤为正例

模型A：预测对了 3 个恶性肿瘤样本，4 个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：3
2.伪反例 FN 为：3
3.伪正例 FP 为：0
4.真反例 TN：4

精度：100%
召回率：50%

F1-score: 67%

模型B：预测对了 6 个恶性肿瘤样本，1个良性肿瘤样本

请计算：TP、FN、FP、TN

1.真正例 TP 为：6
2.伪反例 FN 为：0
3.伪正例 FP 为：3
4.真反例 TN：1

精度： 67%
召回率： 100

F1-score：80%

分类评估方法 - 精确率、召回率、F1-score

- F1-score - 举个栗子 – 程序计算

```
# 1.导入依赖包
from sklearn.metrics import f1_score

def dm03_F1():
    # 2.构建数据，样本集中共有6个恶性肿瘤样本, 4个良性肿瘤样本
    y_true = [“恶性”, “恶性”, “恶性”, “恶性”, “恶性”, “恶性”, “良性”, “良性”, “良性”, “良性”]

    # 3.1 模型F1-score评估，模型A: 预测对了3个恶性肿瘤样本, 4个良性肿瘤样本
    y_pred = [“恶性”, “恶性”, “恶性”, “良性”, “良性”, “良性”, “良性”, “良性”, “良性”, “良性”]
    result = f1_score(y_true, y_pred, pos_label=“恶性”)
    print(“模型Af1-score:”, result)

    # 3.2 模型F1-score评估，模型B: 预测对了6个恶性肿瘤样本, 1个良性肿瘤样本
    y_pred = ["恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "恶性", "良性"]
    result = f1_score(y_true, y_pred, pos_label="恶性")
    print("模型Bf1-score::", result)
```

总结

1、 混淆矩阵的四个指标

真实结果 \ 预测结果	正例	假例
正例	真正例TP	伪反例FN
假例	伪正例FP	真反例TN

3 精确率，精度 Precision $P = \frac{TP}{TP+FP}$

4 召回率，也叫查全率 Recall $P = \frac{TP}{TP+FN}$

5 F1-score

- 综合能力、综合稳定性指标 F1-score
- $P = \frac{2 * Precision * Recall}{Precision + Recall}$

练习

关于精确率、召回率和 F1 值的定义，下列说法正确的是：

A. 精确率是指预测为正例中实际为正例的比例，召回率是指实际为正例中被预测为正例的比例，F1 值是精确率和召回率的调和平均数。

B. 精确率是指实际为正例中被预测为正例的比例，召回率是指预测为正例中实际为正例的比例，F1 值是精确率和召回率的调和平均数。

C. 精确率是指实际为正例中被预测为正例的比例，召回率是指预测为正例中实际为正例的比例，F1 值是精确率和召回率的算术平均数。

答案：A

练习

如果一个二分类模型的精确率为 0.8，召回率为 0.6，那么该模型的 F1 值为：
正确答案：C

A. 0.44

B. 0.60

C. 0.69

D. 0.75

答案：C

目录

Contents

- 逻辑回归简介

  应用场景，数学知识

- 逻辑回归原理

- 逻辑回归API函数和案例

- 分类问题评估

  混淆矩阵、精确率、召回率、F1-score、AUC指标、ROC曲线

- 电信客户流失预测案例

学习目标

Learning Objectives

1. 知道ROC曲线和AUC指标
2. 了解相应的API函数

分类评估方法 -ROC曲线、AUC指标

- **真正率TPR与假正率FPR**

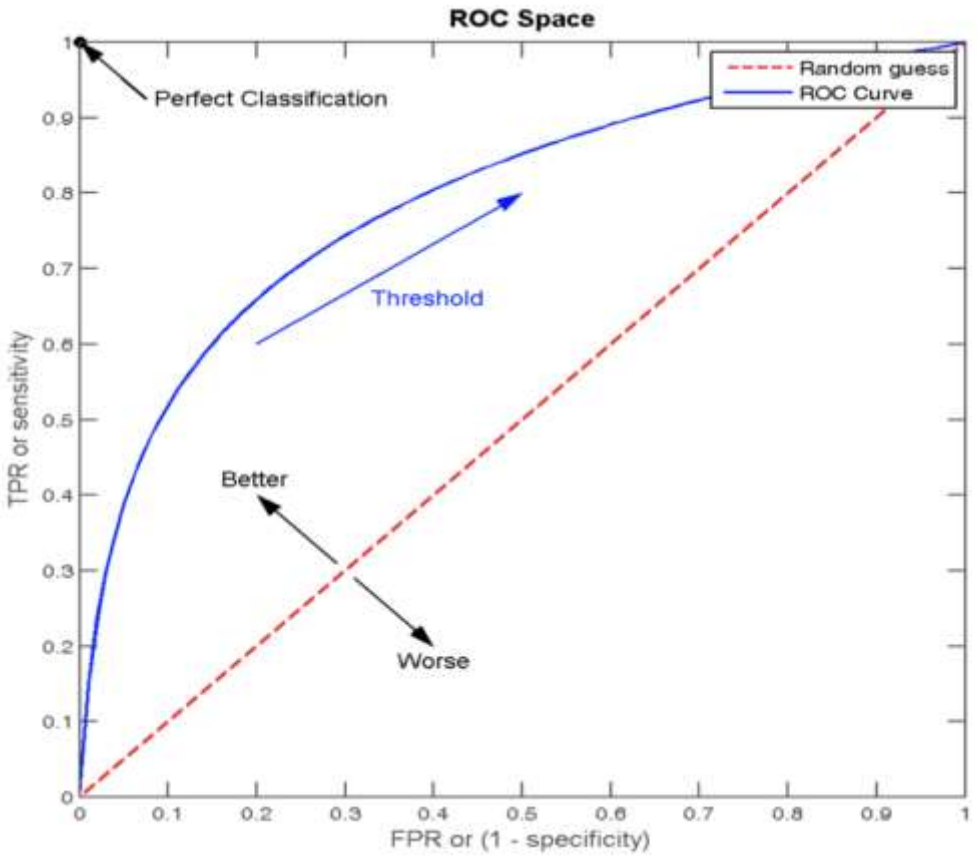
  1 正样本中被预测为正样本的概率TPR （True Positive Rate）

  2 负样本中被预测为正样本的概率FPR （False Positive Rate）

  通过这两个指标可以描述模型对正/负样本的分辨能力

- **ROC曲线（Receiver Operating Characteristic curve）**

  是一种常用于评估分类模型性能的可视化工具。ROC曲线以模型的真正率TPR为纵轴，假正率FPR为横轴，它将模型在不同阈值下的表现以曲线的形式展现出来。



- **AUC (Area Under the Curve） - ROC曲线下面积**

ROC曲线的优劣可以通过曲线下的面积（AUC）来衡量，AUC越大表示分类器性能越好。

当AUC=0.5时，表示分类器的性能等同于随机猜测

当AUC=1时，表示分类器的性能完美，能够完全正确地将正负例分类。

分类评估方法 - ROC曲线、AUC指标

- **ROC 曲线图像中，4 个特殊点的含义**

点坐标说明：图像x轴FPR/y轴TPR, 任意一点坐标A(FPR值，TPR值)

1.点(0, 0)： 所有的负样本都预测正确，所有的正样本都预测错误 。相当于点的(FPR值0，TPR值0)

2.点(1, 0)： 所有的负样本都预测错误，所有的正样本都预测错误。相当于点的(FPR值1，TPR值0)

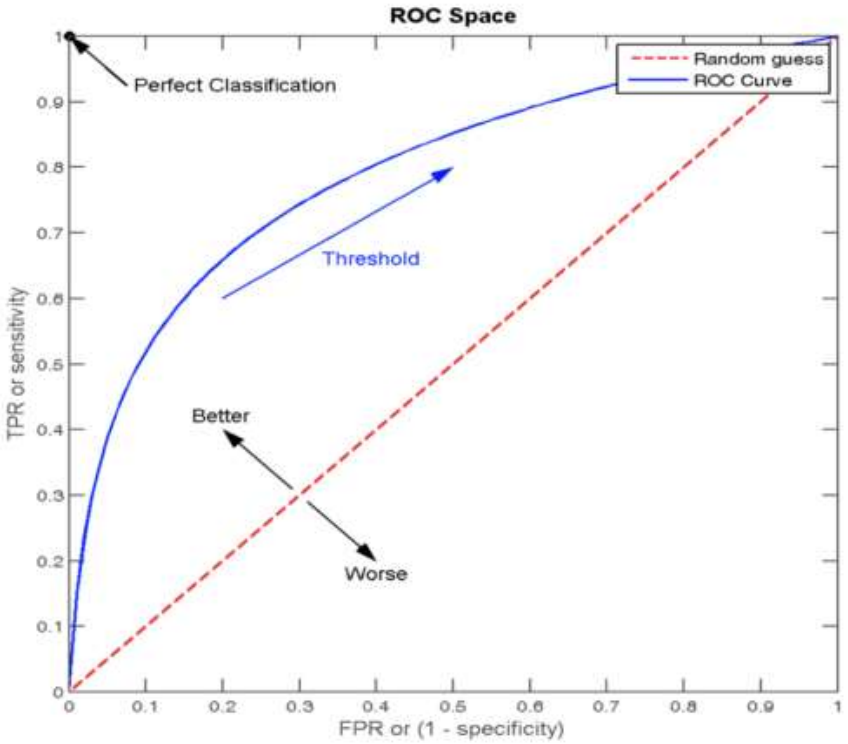
- 最差的效果

1.点(1, 1)：所有的负样本都预测错误，表示所有的正样本都预测正确。相当于点的(FPR值1，TPR值1)

2.点(0, 1)：所有的负样本都预测正确，表示所有的正样本都预测正确 。相当于点的(FPR值0，TPR值1)

- 最好的效果

ROC曲线上的每个点代表模型在不同阈值下的性能表现



- **从图像上来看**

曲线越靠近 (0,1) 点则模型对正负样本的辨别能力就越强

AUC 是 ROC 曲线下面的面积，该值越大，则模型的辨别能力就越强，AUC 范围在 [0, 1] 之间

当 AUC= 1 时，该模型被认为是完美的分类器，但是几乎不存在完美分类器

当 AUC <= 0.5 时，模型区分正负样本的就会变得模棱两可，近似于随机猜测

分类评估方法 - ROC曲线、AUC指标

- **案例：ROC 曲线的绘制**

已知：在网页某个位置有一个广告图片，该广告共被展示了 6 次； 有 2 次被浏览者点击了。每次点击的概率见图1。

其中正样本{1, 3} 负样本为{2, 4, 5, 6}

要求画出：在不同阈值下的ROC曲线。

图1：每次点击的概率图

样本	是否被点击	预测点击概率
1	1	0.9
2	0	0.7
3	1	0.8
4	0	0.6
5	0	0.5
6	0	0.4

图2：根据预测点击概率排序之后的图

样本	是否被点击	预测点击概率
1	1	0.9
3	1	0.8
2	0	0.7
4	0	0.6
5	0	0.5
6	0	0.4

思路分析：根据不同的阈值，求出TPR、FPR，得出点坐标画ROC图

分类评估方法 - ROC曲线、AUC指标

阈值：0.9

原本为正例的 1、3 号的样本中 1、3 号样本全被分类错误，则 TPR = 0/2 = 0

原本为负例的 2、4、5、6 号样本没有一个被分为正例，则 FPR = 0

阈值：0.8

原本为正例的 1、3 号样本中 1 号样本被分类正确，则 TPR = 1/2 = 0.5

原本为负例的 2、4、5、6 号样本没有一个被分为正例，则 FPR = 0

阈值：0.7

原本为正例的 1、3 号样本被全部分类正确，则 TPR = 2/2 = 1

原本为负类的 2、4、5、6 号样本中没有一个被分类错误，则 FPR = 0/4 = 0

阈值：0.6

原本为正例的 1、3 号样本被分类正确，则 TPR = 2/2 = 1

原本为负类的 2、4、5、6 号样本中 2 号样本被分类错误，则 FPR = 1/4 = 0.25

阈值：0.5

原本为正例的 1、3 号样本被分类正确，则 TPR = 2/2 = 1

原本为负类的 2、4、5、6 号样本中 2、4 号样本被分类错误，则 FPR = 2/4 = 0.5

阈值 0.4

原本为正例的 1、3 号样本被分类正确，则 TPR = 2/2 = 1

原本为负类的 2、4、5、6 号样本中2、4、5 号样本被分类错误，则 FPR = 3/4 = 0.75

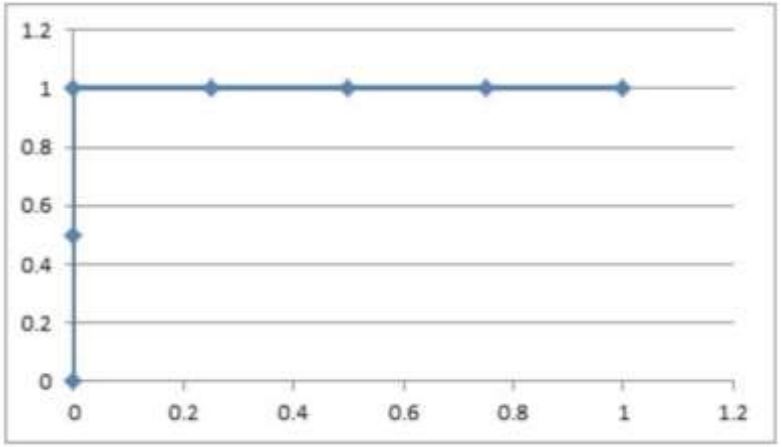
- **案例：ROC 曲线的绘制**

会得出不同阈值下的点坐标：

(0, 0.5)、(0, 1)、(0.25, 1)、

(0.5, 1)、(0.75, 1)、(1, 1)

由 TPR 和 FPR 构成的 ROC 图像为



ROC曲线上的每个点
代表模型在不同阈值下的性能表现

分类评估方法 - ROC曲线、AUC指标

- **AUC的计算API**

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_true, y_score)
```

计算ROC曲线面积，即AUC值

y_true：每个样本的真实类别，必须为0(反例),1(正例)标记

y_score：预测得分，可以是正例的估计概率、置信值或者分类器方法的返回值

- **分类评估报告API**

sklearn.metrics.classification_report(y_true, y_pred, labels=[], target_names=None)

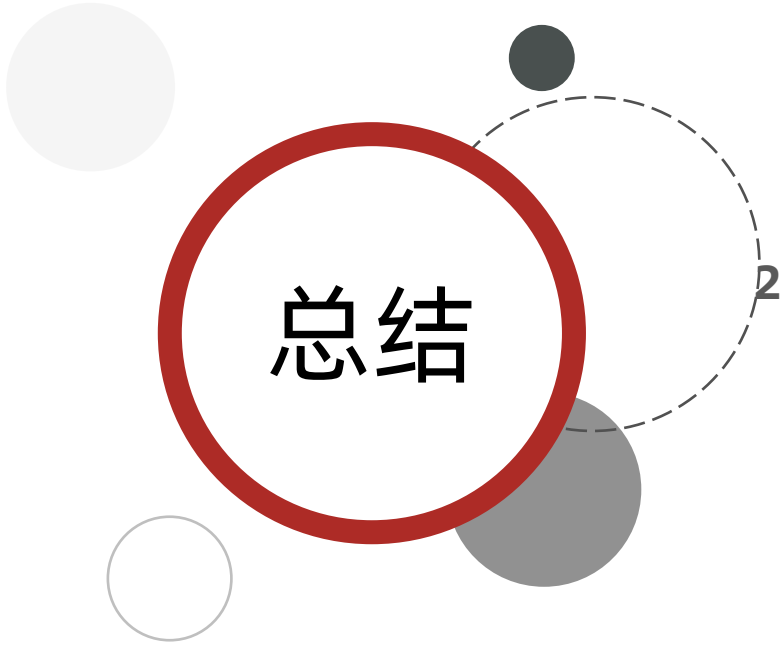
y_true：真实目标值

y_pred：估计器预测目标值

labels:指定类别对应的数字

target_names：目标类别名称

return：每个类别精确率与召回率



1 ROC曲线

ROC曲线以模型的真正率TPR、假正率FPR为y/x轴，它将模型在不同阈值下的表现以曲线的形式展现出来。

2 AUC指标

- AUC越大表示分类器性能越好
- 当AUC=0.5时，表示分类器的性能等同于随机猜测
- 当AUC=1时，表示分类器的性能完美，能够完全正确地将正负例分类

练习

1、下列关于逻辑回归模型的评估说法正确的是？（**多选**）

A）我们在评估逻辑回归模型时只需要选择一种评估方法即可

B）混淆矩阵能够帮助我们快速计算出其它分类模型指标

C）召回率和精确率表达的是同样的概念

D）ROC曲线下与坐标轴形成的闭合区域的面积即为AUC指标的值

答案解析：BD

练习

1 以下哪个描述最准确地解释了AUC指标的含义？

A. AUC是ROC曲线下方的面积，用于比较分类器的性能。

B. AUC是ROC曲线上某一点的斜率，表示分类器的灵敏度。

C. AUC是ROC曲线上的最大误分类率，用于评估分类器的错误率。

D. AUC是ROC曲线上的阈值，用于选择最佳分类器。

答案：A

2 下面哪个选项最能说明ROC曲线的作用？

A. ROC曲线用于描述模型的参数。

B. ROC曲线用于可视化模型的损失函数。

C. ROC曲线用于评估二元分类模型的性能。

D. ROC曲线用于选择最佳的回归模型。

答案：C

目录

Contents

学习目标

Learning Objectives

1. 了解案例的背景信息
2. 知道案例的处理流程
3. 动手实现电信客户流失案例的代码

案例 -电信客户流失预测

- 案例需求
  - 已知：用户个人，通话，上网等信息数据
  - 需求：通过分析特征属性确定用户流失的原因，以及哪些因素可能导致用户流失。建立预测模型来判断用户是否流失，并提出用户流失预警策略。
- 数据集介绍
  - CustomerID 客户ID
  - Gender 性别
  - partneratt 配偶是否也为att用户
  - dependents_att 家人是否也是att用户
  - landline 是否使用att固话服务
  - internet_att/internet_other 是否使用att的互联网服务
  - Paymentbank/creditcard/electroinc 付款方式
  - MonthlyCharges 每月话费
  - TotalCharges 累计话费
  - Contract_month/1year 用户使用月度/年度合约
  - StreamingTv/streamingMovies 是否使用在线视频或者电影app
  - Churn 客户转化的flag

案例 -电信客户流失预测

	A	B	C	D	E	F	G	Strear
1	Churn	gender	Partner_att	Dependents_att	landline	internet_att	internet_other	
2	No	Female	1	0	0	1	0	
3	No	Male	0	0	1	1	0	
4	Yes	Male	0	0	1	1	0	
5	No	Male	0	0	0	1	0	
6	Yes	Female	0	0	1	0	1	
7	Yes	Female	0	0	1	0	1	
8	No	Male	0	1	1	0	1	
9	No	Female	0	0	0	1	0	
10	Yes	Female	1	0	1	0	1	

	L	M	N	O	P
1	PaymentBank	PaymentCreditcard	PaymentElectronic	MonthlyCharges	TotalCharges
2	0	0	1	29.85	29.85
3	0	0	0	56.95	1889.5
4	0	0	0	53.85	108.15
5	1	0	0	42.3	1840.75
6	0	0	1	70.7	151.65
7	0	0	1	99.65	820.5
8	0	1	0	89.1	1949.4
9	0	0	0	29.75	301.9
10	0	0	1	104.8	3046.05

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 16 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Churn              7043 non-null   object
 1   gender             7043 non-null   object
 2   Partner_att        7043 non-null   int64
 3   Dependents_att     7043 non-null   int64
 4   landline           7043 non-null   int64
 5   internet_att       7043 non-null   int64
 6   internet_other     7043 non-null   int64
 7   StreamingTV        7043 non-null   int64
 8   StreamingMovies    7043 non-null   int64
 9   Contract_Month     7043 non-null   int64
 10  Contract_1YR       7043 non-null   int64
 11  PaymentBank        7043 non-null   int64
 12  PaymentCreditcard  7043 non-null   int64
 13  PaymentElectronic  7043 non-null   int64
 14  MonthlyCharges     7043 non-null   float64
 15  TotalCharges       7043 non-null   float64
dtypes: float64(2), int64(12), object(2)
memory usage: 880.5+ KB
```

案例 -电信客户流失预测

- 案例步骤分析

1、数据基本处理

- 主要是查看数据行/列数量
- 对类别数据数据进行one-hot处理
- 查看标签分布情况

2、特征筛选(特征工程)

- 分析哪些特征对标签值影响大
- 对标签进行分组统计，对比0/1标签分组后的均值等
- 初步筛选出对标签影响比较大的特征，形成x、y

3、模型训练

- 样本均衡情况下模型训练
- 样本不平衡情况下模型训练
- 交叉验证网格搜素等方式模型训练

4、模型评估

- 精确率
- ROC_AUC指标计算

案例 -电信客户流失预测 - 1数据基本处理

```
import numpy as np
import pandas as pd

def dm01_数据基本处理():
    churn_pd = pd.read_csv('./data/churn.csv')
    print(f'data.info-->{data.info}')
    print('churn_pd.describe()-->', churn_pd.describe())
    print('churn_pd-->', churn_pd)

    # 1 处理类别型的数据 类别型数据做one-hot编码
    churn_pd = pd.get_dummies(churn_pd)
    print('churn_pd-->', churn_pd)
    print(churn_pd.info)

    # 2 去除列Churn_no gender_Male
    churn_pd  = churn_pd.drop(['Churn_No', 'gender_Male'], axis=1)
    print(churn_pd.info)

    # 3 列标签重命名 打印列名
    print('churn_pd.columns', churn_pd.columns)
    churn_pd  = churn_pd.rename(columns = {'Churn_Yes':'flag'})
    print('churn_pd.columns', churn_pd.columns)

    # 4 查看标签的分布情况 0.26用户流失
    value_counts = churn_pd.flag.value_counts(1)
    print('value_counts-->\n', value_counts)
    print('从标签的分类中可以看出: 属于标签分类不平衡样本')
```

案例 -电信客户流失预测 - 特征筛选

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

def dm02_特征筛选():
    churn_pd = pd.read_csv('./data/churn.csv')
    # 1 处理类别型的数据 类别型数据做one-hot编码
    churn_pd = pd.get_dummies(churn_pd)

    # 2 去除列 Churn_no gender_Male # nplace=True 在原来的数据上进行删除
    churn_pd.drop(['Churn_No', 'gender_Male'], axis=1, inplace=True)

    # 3 列标签重命名 打印列名
    churn_pd.rename(columns={'Churn_Yes': 'flag'}, inplace=True)

    # 4 查看标签的分布情况 0.26用户流失
    value_counts = churn_pd.flag.value_counts(1)


    # 5 查看Contract_Month 是否月签约流失情况
    sns.countplot(data=churn_pd, y = "Contract_Month", hue='flag')
    plt.show()
```

案例 -电信客户流失预测 - 3 模型训练与评测

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score
from sklearn.metrics import classification_report
import pandas as pd

def dm03_模型训练和评测():
    # 1 数据基本处理
    churn_pd = pd.read_csv('./data/churn.csv')
    # 1-1 处理类别型的数据 类别型数据做one-hot编码
    churn_pd = pd.get_dummies(churn_pd)

    # 1-2 去除列Churn_no gender_Male # nplace=True 在原来的数据上进行删除
    churn_pd.drop(['Churn_No', 'gender_Male'], axis=1, inplace=True)

    # 1-3 列标签重命名 打印列名
    churn_pd.rename(columns={'Churn_Yes': 'flag'}, inplace=True)

    # 2 特征处理
    # 2-1 确定目标值和特征值
    x = churn_pd[['Contract_Month', 'internet_other', 'PaymentElectronic']]
    y = churn_pd['flag']

    # 2-2 数据集划分
    x_train, x_test, y_train, y_test =  train_test_split(x, y, test_size=0.3,
random_state=100)
```

```
    # 3 实例化模型 训练模型 模型预测
    estimator =  LogisticRegression()
    estimator.fit(x_train, y_train)
    y_pred = estimator.predict(x_test)

    # 4 模型评估
    my_accuracy_score = accuracy_score(y_test, y_pred)
    print('my_accuracy_score-->', my_accuracy_score)

    my_score = estimator.score(x_test, y_test)
    print('my_score-->', my_score)

    # 计算AUC值
    my_roc_auc_score = roc_auc_score(y_test, y_pred)
    print('my_roc_auc_score-->', my_roc_auc_score)

    result = classification_report(y_test, y_pred, target_names=['flag0', 'flag1'])
    print('classification_report result->\n', result)
```


黑马程序员
www.itheima.com
传智教育旗下高端IT教育品牌