多元统计与矩阵分析

张锋　*8125345@qq.com*
中国地质大学, 计算机学院, 武汉

第10章
逻辑回归模型

内容

(1) 基本思想
(2) 逻辑回归的数学模型
(3) 逻辑回归模型的参数估计
(4) 逻辑回归模型的模型检验

基本思想

- 一种广义线性模型
- 实质是一种分类模型
- 响应变量的取值范围仅为1和0，代表响应变量的两种情况发生和不发生

线性回归 + logistic function = logistic回归

基本思想

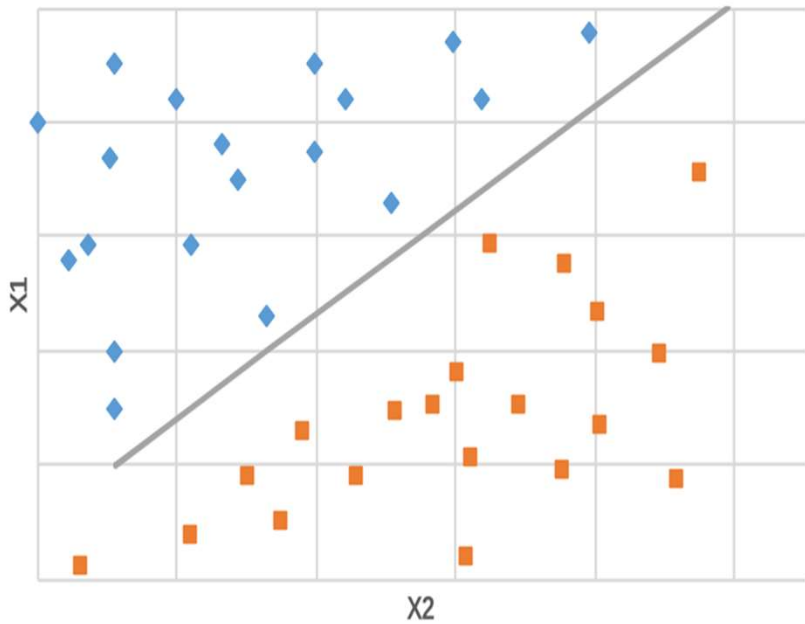
- 一般线性回归模型的形式为：

$$y = \alpha + \beta x$$

  - 自变量x的取值范围为$(-\infty, +\infty)$，响应变量y的取值范围也为$(-\infty, +\infty)$。
  - 但是逻辑回归中，响应变量y的取值范围仅为0和1两种情况，而自变量x的取值范围仍为$(-\infty, +\infty)$

基本思想

- 有两类样本“是”和“否”，分别用图中的蓝点和黄点表示。x_1和x_2分别为两个变量。
- 目标是：利用已知分类结果的样本信息，寻找两类结果的判断规则，尽量将样品分开；并利用这一规则，对分类结果未知的样品进行判断



- 决策边界(Decision Bound)：$\theta_0 + \theta_1 x_1 + \theta_2 x_2$

- 推广到高维空间，决策边界为一个分类超平面（Hyperplane ）：

$$\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

基本思想

- 以信用卡评分为例，我们可选择用户的年龄、性别、收入水平等作为自变量，因变量则可定义为是否有过逾期行为。

内容

(1) 基本思想

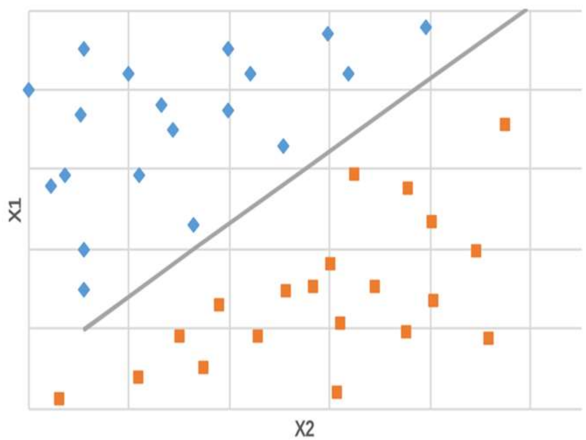
(2) 逻辑回归的数学模型

(3) 逻辑回归模型的参数估计

(4) 逻辑回归模型的模型检验

逻辑回归实质是分类模型

- Logistic Regression 虽然被称为回归，但其实际上是分类模型，并常用于二分类。Logistic Regression 因其简单、可并行化、可解释强深受工业界喜爱。
- Logistic 回归的本质是：假设数据服从这个分布，然后使用极大似然估计做参数的估计。



逻辑回归实质是分类模型

- 逻辑回归的决策边界为一个分类超平面（Hyperplane ）：

$$\alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- “分类”是应用逻辑回归的目的和结果，但中间过程依旧是“回归”。

逻辑回归实质是分类模型

- 逻辑回归函数（Sigmod）函数

$$f(y) = \frac{1}{1+e^{-y}}$$

- 定义：$\ln\frac{p}{1-p} = \alpha + \beta x$

- 因此可得逻辑回归的数学形式为：

$$p = \frac{1}{1+e^{-(\alpha+\beta x)}}$$

Logit变换与Logistic回归

- 优势($Odds$)

$$odds = \frac{p}{1-p}$$

- logit变换

$$\ln(odds) = \ln\frac{p}{1-p}$$

- logit模型

$$\ln(odds) = \ln\frac{p}{1-p} = \alpha + \beta x$$

模型解释

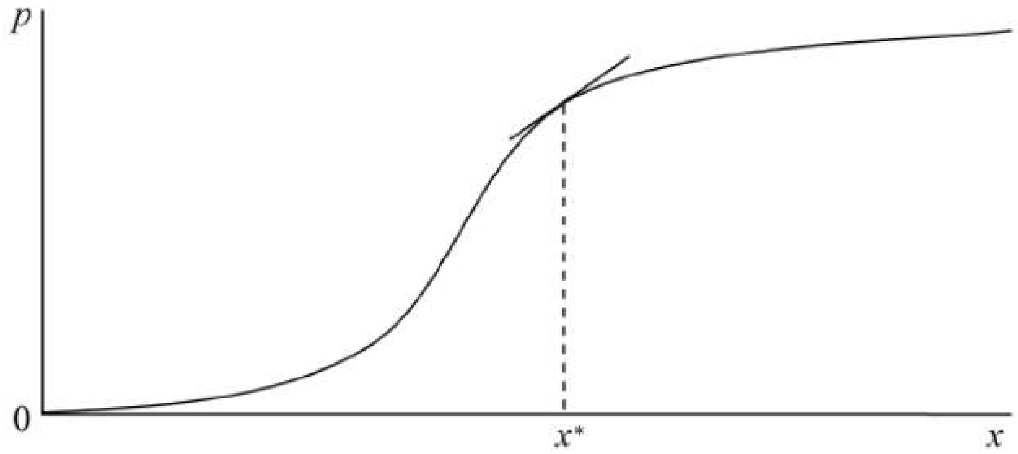


图 10-2　$\beta > 0$ 时的一元 Logistic 回归曲线示意

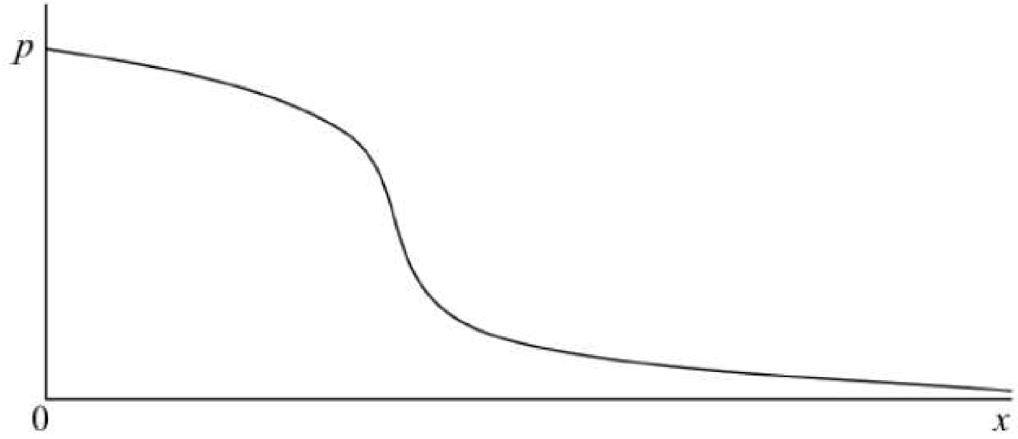


图 10-3　$\beta < 0$ 时的一元 Logistic 回归曲线示意

内容

(1) 基本思想

(2) 逻辑回归的数学模型

(3) 逻辑回归模型的参数估计

(4) 逻辑回归模型的模型检验

Logistic回归的参数估计

最大似然估计法

逻辑回归的似然函数为：

$$L(\beta)=\prod_{i=1}^{n}P(y=1|x_i;\beta)^{y_i}[1-P(y=1|x_i;\beta)]^{1-y_i}$$

$$=\prod_{i=1}^{n}\left(\frac{1}{1+e^{-(\alpha+\beta x_i)}}\right)^{y_i}\left(1-\frac{1}{1+e^{-(\alpha+\beta x_i)}}\right)^{1-y_i}$$

对数似然函数为：

$$\ln L(\beta)=\sum_{i=1}^{n}\left\{y_i\times\ln\frac{1}{1+e^{-(\alpha+\beta x_i)}}+\left[\left(1-y_i\right)\times\ln(1-\frac{1}{1+e^{-(\alpha+\beta x_i)}}\right)\right]\right\}$$

$$=\sum_{i=1}^{n}\left[y_i(\alpha+\beta x_i)-\ln(1+e^{\alpha+\beta x_i})\right]$$

关于参数β的偏导，可得到似然方程：

$$\frac{\partial \ln L(\beta)}{\partial(\beta)}=\sum_{i=1}^{n}\left(y_i x_i-\frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}}x_i\right)=0$$

陷入循环求解的过程

解得：

$$y_i=\frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}}$$

等价于

$$p=\frac{1}{1+e^{-(\alpha+\beta x)}}$$

如何求解？最小二乘估计可行吗？

Logistic回归的参数估计

解决这一问题的思路是，通过求解$-\ln L(\theta)$的极小值来进行估计，即设定变换函数$J(\theta)$，有：

$$J(\beta)=-\ln L(\beta)$$

$$=-\sum_{i=1}^{n}\left\{y_i\times\ln\frac{1}{1+e^{-(\alpha+\beta x_i)}}+\left[\left(1-y_i\right)\times\ln\left(1-\frac{1}{1+e^{e^{-(\alpha+\beta x_i)}}}\right)\right]\right\}$$

凸优化中的梯度下降法、牛顿法、拟牛顿法均可以求解！

梯度下降法

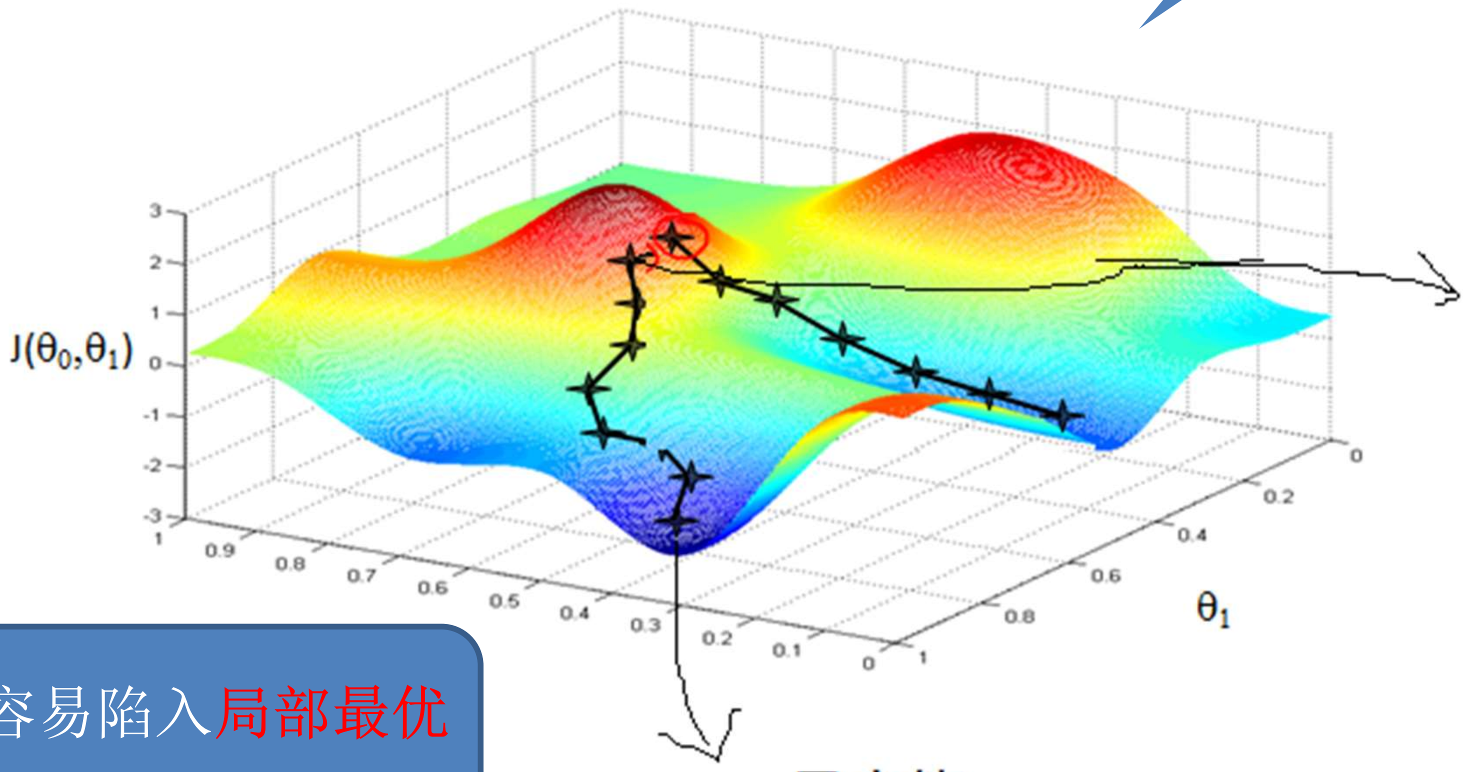
- 通过求解目标函数$f(x)$在某取值处的一阶偏导数，从而确定在特定的位置，如$(x_0, y_0, \cdots, z_0)$处的梯度；
- 由于$f(x)$在该点的梯度方向是函数值增长最快的方向，所以其反方向则是函数值下降最快的方向

$$\beta_{k+1} = \beta_k - a\nabla J(\beta)$$

β_{k+1}、β_k为分别为参数θ在$k+1$、k时刻的取值；α为梯度下降的迭代过程的步长；$\nabla J(\beta)$为梯度的方向，$-\nabla J(\beta)$为梯度的反方向。

若目标函数为凸函数，则凸函数的性质可知，局部最优为全局最优

著名的下山问题：



局限：容易陷入局部最优的困境

以函数 $f(x) = x^2$ 为例，其梯度为 $f'(x) = 2x$，初始化时，我们令起点$x_0 = 1$，步长$\alpha = 0.4$，根据梯度下降的公式可得：

$$
\begin{aligned}
x_0 &= 1 \\
x_1 &= x_0 - \alpha f'(x_0) \\
&= 1 - 0.4 * 2 \\
&= 0.2 \\
x_2 &= x_1 - \alpha f'(x_1) \\
&= 0.04 \\
x_3 &= 0.008 \\
x_4 &= 0.0016
\end{aligned}
$$

可知第三次和第四次迭代得到的估计值比较接近，故经过四次运算，我们近似可以得到x的值了。

那么，逻辑回归的梯度则可表示为：

$$\nabla J(\beta) = -\sum_{i=1}^{n}\left(y_i x_i - \frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}} x_i\right)$$

?

其参数的梯度下降法表示为：

$$\beta_{k+1} = \beta_k - a\left[-\sum_{i=1}^{n}\left(y_i x_i - \frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}} x_i\right)\right]$$

$$= \beta_k - a\left[\sum_{i=1}^{n}\left(\frac{e^{\alpha+\beta x_i}}{1+e^{\alpha+\beta x_i}} x_i - y_i x_i\right)\right]$$

不断迭代，当两次迭代的结果比较接近时，即可求得逻辑回归参数的近似解

那么，逻辑回归的梯度则可表示为：

$$\nabla J(\beta) = -\sum_{i=1}^{n}\left(y_i\boldsymbol{x}_i - \frac{e^{\boldsymbol{\beta}\boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}\boldsymbol{x}_i}}\boldsymbol{x}_i\right)$$

其参数的梯度下降法表示为：

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - a\left[-\sum_{i=1}^{n}\left(y_i\boldsymbol{x}_i - \frac{e^{\boldsymbol{\beta}_i\boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}\boldsymbol{x}_i}}\boldsymbol{x}_i\right)\right]$$

$$= \boldsymbol{\beta}_k - a\left[\sum_{i=1}^{n}\left(\frac{e^{\boldsymbol{\beta}\boldsymbol{x}_i}}{1+e^{\boldsymbol{\beta}\boldsymbol{x}_i}}\boldsymbol{x}_i - y_i\boldsymbol{x}_i\right)\right]$$

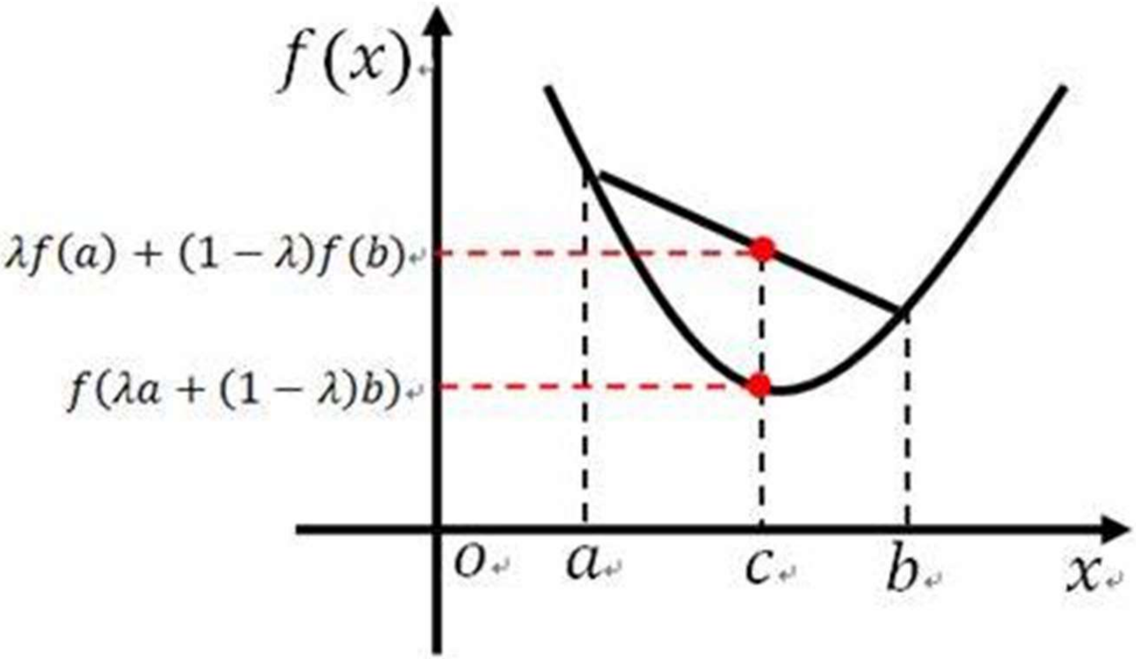
不断迭代，当两次迭代的结果比较接近时，即可求得逻辑回归参数的近似解

然而这样求得的最优值有可能仅为“局部最优”，当目标函数为凸函数时，根据图函数的性质可以知道局部最优必为全局最优。因此，求得目标函数$J(\theta)$的二阶偏导数为：

$$\frac{\partial^2 J(\beta)}{\partial \beta^2} = -\sum_{i=1}^{n}\left[-\frac{e^{\alpha+\beta x_i}}{(1+e^{\alpha+\beta x_i})^2}x_i\right] = \sum_{i=1}^{n}\left[\frac{e^{\alpha+\beta x_i}}{(1+e^{\alpha+\beta x_i})^2}x_i\right]$$

由于$x_i^2 \geq 0$，$e^{\alpha+\beta x_i} \geq 0$，$\left(1+e^{\alpha+\beta x_i}\right)^2 \geq 0$，故$\frac{\partial^2 J(\theta)}{\partial \theta^2} \geq 0$

根据凸函数的充要条件可知，$J(\theta)$为凸函数，故局部最优点必为全局最优点，从而可知，所求的参数估计值即为我们所求的参数估计值。



通过函数图形，我们非常直观的就能看到，如果一个函数是凸函数的话，其局部最小值就是全局最小值。

内容

(1) 基本思想

(2) 逻辑回归的数学模型

(3) 逻辑回归模型的参数估计

(4) 逻辑回归模型的模型检验

逻辑回归模型中的检验

- 对回归系数的显著性检验
  - *Wald*检验(*Wald Test*)
  - 似然比检验(*Likelihood Ratio Test, LRT*)
  - 比分检验(*Score Test*)等
- 对模型拟合效果的检验
  - 拟合优度检验(*Test Of Goodness of Fit*)
  - 残差检验(*Residuals Test*)等
  - 检验分类模型的相关指标：
    - 混淆矩阵(*Confusion Matrix*)，
    - 准确率(*Accuracy Rate*)
    - AUC等指标。

回归系数的检验

- 给定模型$\mathrm{logit}(p)=\theta_0+\theta_1x_1+\theta_2x_2$，现需要验证参数$\theta_2$是否显著，
  - *Wald*检验(*Wald Test*)
  - 似然比检验(*Likelihood Ratio Test, LRT*)
  - 比分检验(*Score Test*)
- 记无约束模型（*Full Model*）为：
  - $\mathrm{logit}(p)=\theta_0+\theta_1x_1+\theta_2x_2$
- 记约束模型（*Reduced Model*）为：
  - $\mathrm{logit}(p)=\theta_0+\theta_1x_1$
- 则参数θ_2是否显著对应假设检验为：
  - $H_0\colon\theta_2=0$
  - $H_1\colon\theta_2\neq0$

Wald检验

- $Wald$检验的原理是通过构造无约束估计量与检验值之差的函数，并采用估计标准误差进行归一化处理，验证参数是否具有统计显著性。$Wald$统计量可表示为

$$W_{\beta_j} = \left(\frac{\hat{\beta}_j - \beta_j^0}{SE(\hat{\beta}_j)} \right)^2$$

- $\hat{\beta}_j$、β_j^0分别表示参数β_j的估计值（也即无约束模型中参数θ_2的估计值）和检验值（此处为0）；$SE(\hat{\beta}_j)$为β_j的估计标准误差。统计量W服从自由度为n的卡方分布，即$W \sim \chi^2(n)$

Wald检验的不足

- 当变量的回归系数的绝对值很大时，其估计标准误差有可能膨胀，从而导致统计值变小，提高了犯第二类错误的概率；
- 在样本量较小的情形下，估计标准误差将会直接影响检验结论。
- 如果存在以上两种情况，则Wald检验法的适用性受到一定的影响。

Score检验

- *Score*检验原理是在约束模型中构造关于待检验参数β_j的得分函数(*score function*)，也即一阶偏导数；由最大似然估计的性质可知，在最大似然估计值处，得分函数为0。若在待检验参数β_j处的得分函数显著不为0，则有理由认为原假设不成立。*Score*统计量的计算公式为

$$Score = \frac{\left[u(\beta_j)\right]^2}{I(\beta_j)}$$

- β_j为待检验参数，$u(\beta_j)$为约束模型对数似然函数在待检验参数β_j处的一阶偏导数；$I(\beta_j)$为对应的*Fisher*信息(*Fisher Information*)。
- 在大样本情况下，*score*统计量近似服从正态分布。

Score检验

费舍尔信息的计算公式：

$$I(\beta) = -E_\theta[l''(x|\beta)]$$

则参数 β_2 是否显著对应假设检验为：

$$H_0: \beta_2 = 0$$
$$H_1: \beta_2 \neq 0$$

约束模型（*Reduced Model*）为：

$$\text{logit}(p) = \beta_0 + \beta_1 x_1$$

在本例中，对应的score统计量为：

$$Score = \frac{[u(\beta_2)]^2}{I(\beta_2)}$$

将*Score*统计量与正态分布对应的临界值比较，即可判断原假设是否成立，从而可知变量是否显著。

似然比检验

- 参数β_j是否显著对应假设检验为

$$H_0: \beta_j = 0$$
$$H_1: \beta_j \neq 0$$

- 似然比检验的统计量为

$$LR = -2\log\frac{l(\beta_j|H_0)}{l(\beta_j|H_1)} \sim \chi^2(n)$$

  - $l(\beta_j|H_0)$、$l(\beta_j|H_1)$分别表示约束模型、无约束模型所对应的极大似然函数值，n为自由度。

三种检验的对比

- 相同之处
  - 这三种方法都是在对数似然函数的基础上展开的。
- 不同之处
  - *Wald*检验只需要构造无约束模型
  - *Score*检验只需要构造约束模型
  - 似然比检验则需要构造无约束模型和约束模型

模型拟合效果检验

- 在构建好模型后，我们需要对模型整体进行检验，需要检验模型的拟合能力，或者说对于正负样本的区分度。
- 常用的方法
  - 皮尔逊卡方拟合优度检验
  - 混淆矩阵
  - ROC曲线
  - KS值。

皮尔逊卡方拟合优度检验

- 利用样品i的观测值y_i，可以得到该样品的皮尔逊残差(*Pearson residuals*)，有：

$$\gamma_i = \frac{y_i - E(y_i|x)}{\sqrt{\mathrm{var}(y_i|x)}} = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1-\hat{y}_i)}} \qquad \hat{y}_i = \frac{1}{1+e^{-(\alpha+\beta x_i)}}$$

- 构造皮尔逊卡方拟合优度统计量：

$$\chi^2 = \sum_{i=1}^{n} \gamma_i^2$$

  - χ^2统计量服从自由度为$n-2$的卡方分布。对于多元逻辑回归模型则有自由度$n-(k+1)$，其中，k为解释变量的个数。

混淆矩阵

真实结果	预测结果	
	发生	不发生
发生	TP	FN
不发生	FP	TN

- TP（True Positive）表示“实际情况为发生，预测结果也为发生”的样品数量；
- TN（True Negative）表示“实际情况为不发生，预测结果也为不发生”的样品数量；
- 以上两类均表示实际情况与预测结果相符。
- FN（False Negative）
- FP（False Positive）表示实际情况与预测结果不相符的样品数量。

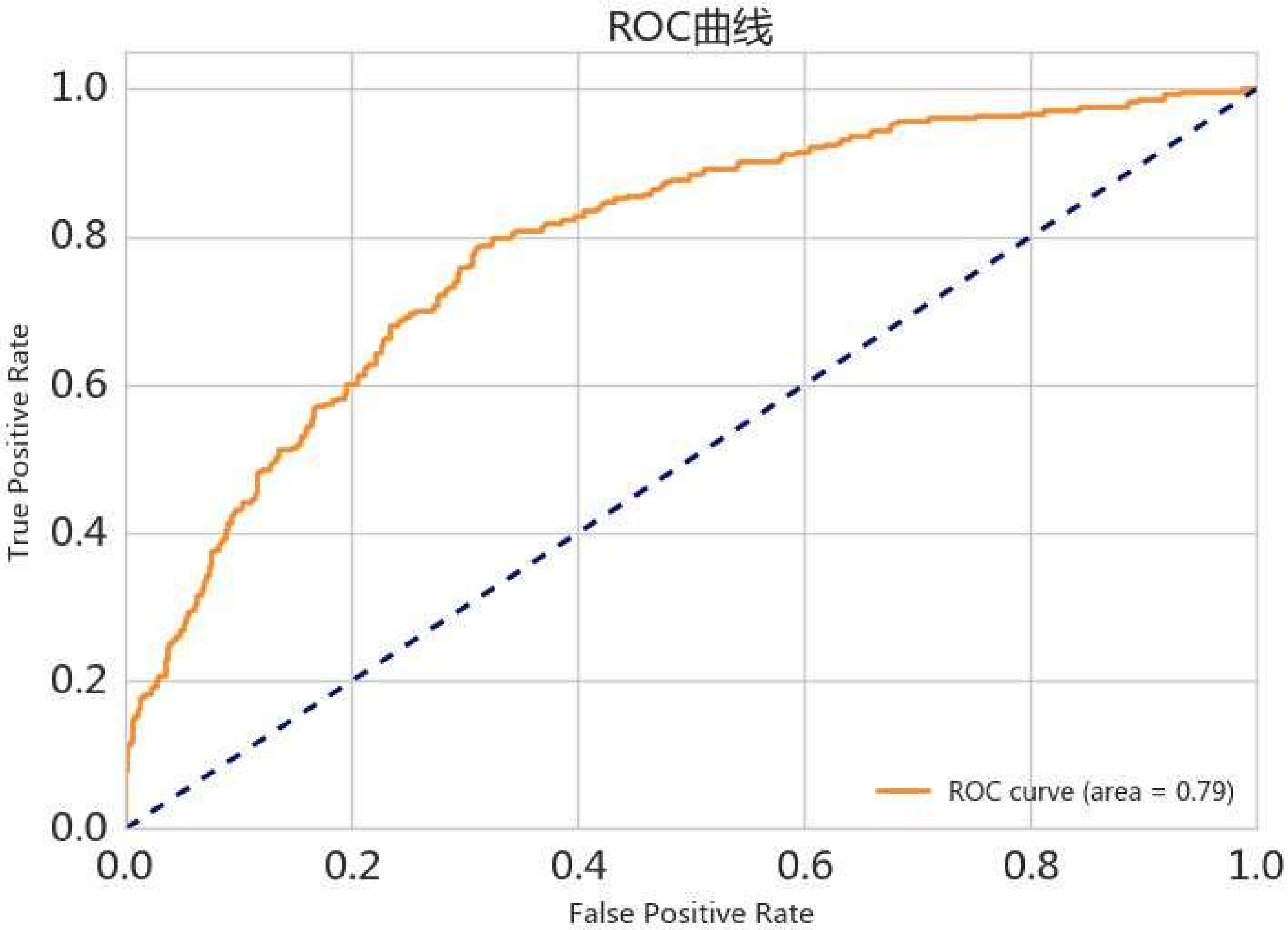
混淆矩阵

- 准确率衡量了正确分类的样品数与样品总数之比$Accu = \frac{TP+TN}{TP+FP+TN+FN}$
- 错误率：$Error = 1 - Accu = \frac{FN+FP}{TP+FP+TN+FN}$
- 精确率表示在预测为“发生”的样品中，有多大比例的样品实际分类就是“发生”的，即：$Precision = \frac{TP}{TP+FP}$
- 真正率（召回率、灵敏度）表示在实际分类为“发生”的样品中有多大比例的样本被预测为“发生”，$TPR = \frac{TP}{TP+FN}$
- 真负率（特异度）表示实际分类为“不发生”的样品中，有多大比例的样本被预测为“不发生”，$TNR = \frac{TN}{FP+TN}$
- 假正率表示实际分类为“不发生”的样品中，有多大比例的样本被预测为“发生”，$FPR = \frac{FP}{FP+TN}$

ROC曲线

- 以TPR为纵坐标，FPR为横坐标，构建一个二维直角坐标系，并在该坐标系中画出点P_i的位置，连成线之后，就得到了ROC曲线(Receiver Operating Characteristic Curve)。
- ROC曲线与横坐标轴围成的面积称为AUC(Area Under Curve of ROC)值，衡量了分类模型的好坏，AUC值越大，则说明分类模型效果越好；反之，不好。

ROC曲线


ROC曲线
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.79)

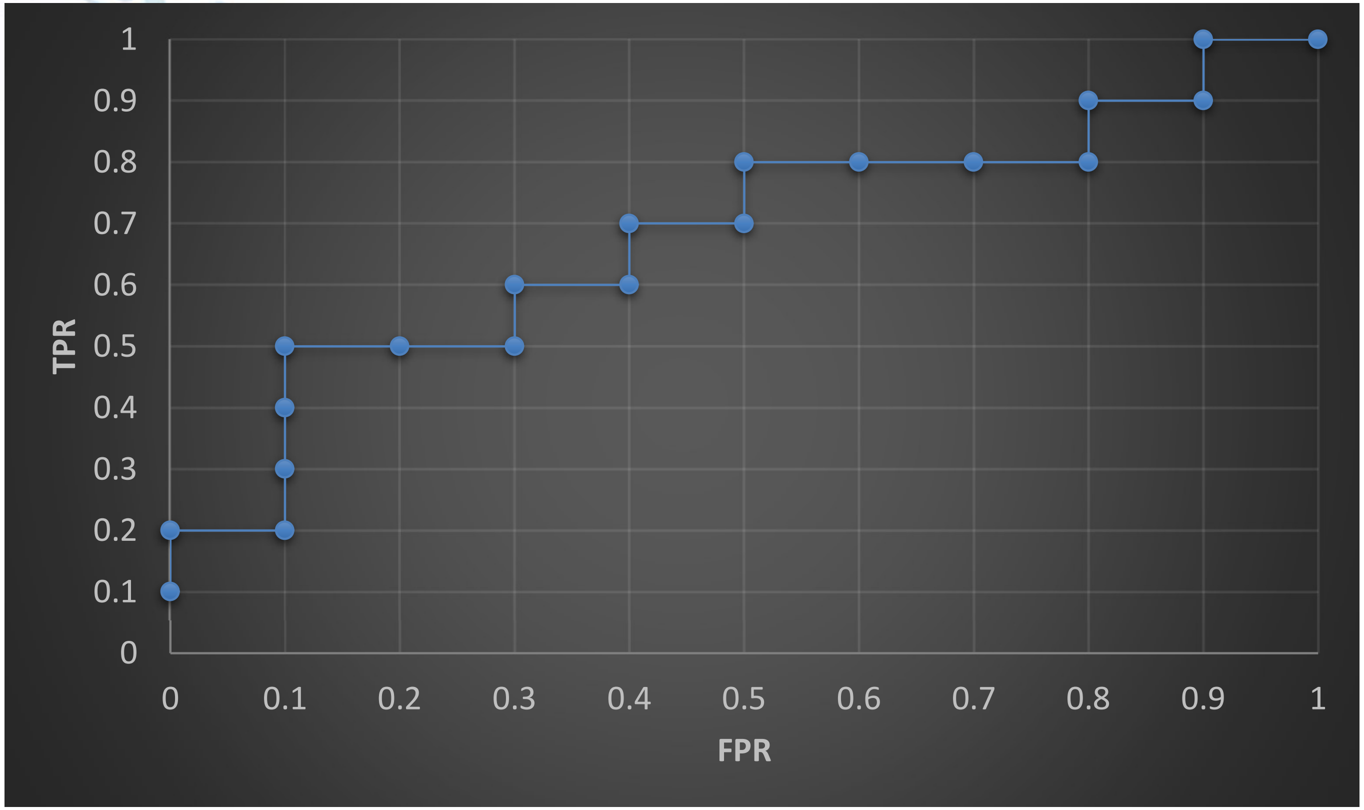
例如，下表给出了20个样本的实际类别以及预测为正类的概率，并将其按照概率进行降序排列：

编号	实际类别	概率	编号	实际类别	概率
1	p	0.9	11	p	0.4
2	p	0.8	12	n	0.39
3	n	0.7	13	p	0.38
4	p	0.6	14	n	0.37
5	p	0.55	15	n	0.36
6	p	0.54	16	n	0.35
7	n	0.53	17	p	0.34
8	n	0.52	18	n	0.33
9	p	0.51	19	p	0.30
10	n	0.505	20	n	0.1

然后依次取不同的概率值为阈值，如先取0.9为阈值，大于等于0.9的记为正样本，小于0.9的负样本，于是便可得到对应的混淆矩阵为：

实际值	预测值	
	正例	负例
正例	1	9
负例	0	10

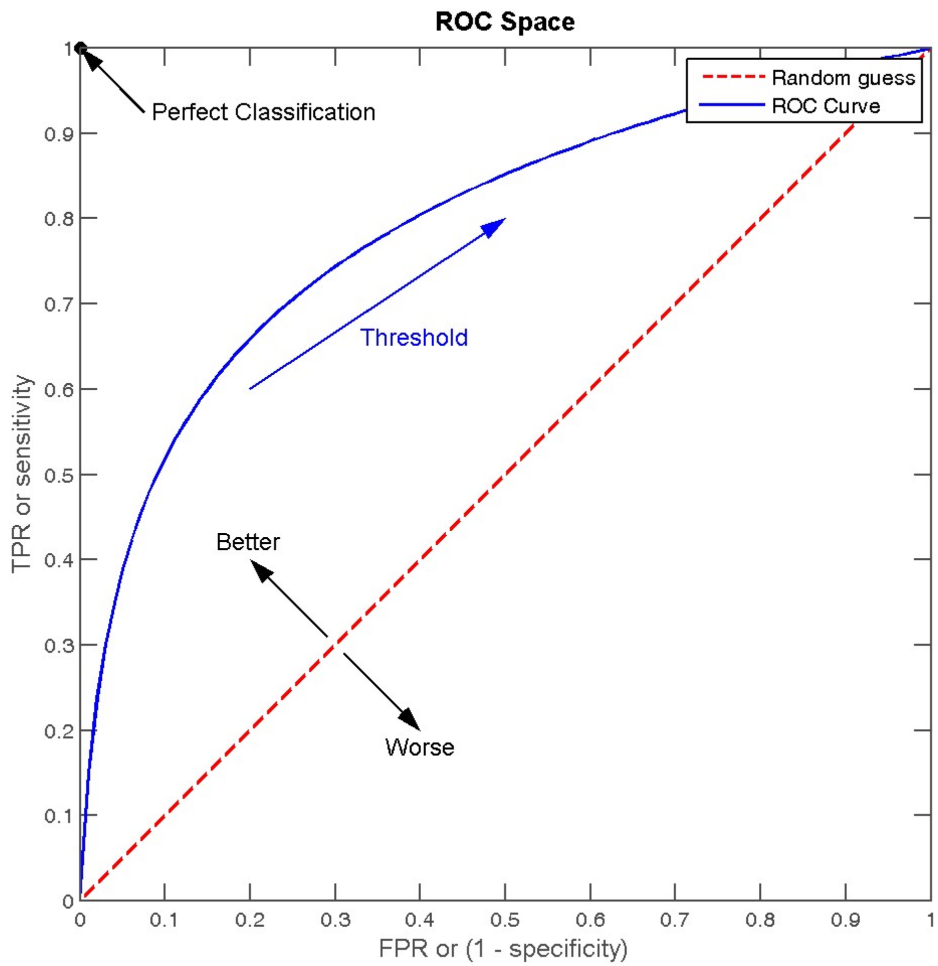
故TPR=1/10，1-TNR=1-10/10，则我们可以得到第一个点的坐标(0, 0.1)；同理，再依次取0.8,0.7, ... ,0.1为阈值，同样也能得到对应的坐标，然后将这些点连起来，便得到了如下的ROC曲线：



AUC为0.68

AUC取值范围	模型效果
=0.5	完全无效果
(0.5,0.6]	模型效果较低
(0.6,0.75]	模型效果中等
(0.75,0.9]	模型效果较高
(0.9,1]	模型效果极高

一个区分能力较好的分类模型，对应的ROC曲线应该是这样子，或者更靠近左上角



KS值

- 来源于Kolmogorov-Smirnov检验，基于累计分布函数
- 用以检验一个经验分布是否符合某种理论分布或者比较两个经验分布是否有显著性差异。
- 在逻辑回归中，取正样本和负样本两个分布的累计分布函数做差值的绝对值运算，然后取其中最大的值作为KS值。
- 给定KS值的参照表，我们便可对模型的拟合效果做出评价。

KS值	模型的对正负样本的区分度
<=0.2	无
(0.2,0.4]	低
(0.4,0.5]	中
(0.5,0.6]	高
(0.6,0.75]	极高
>0.75	太高了，模型可能有误

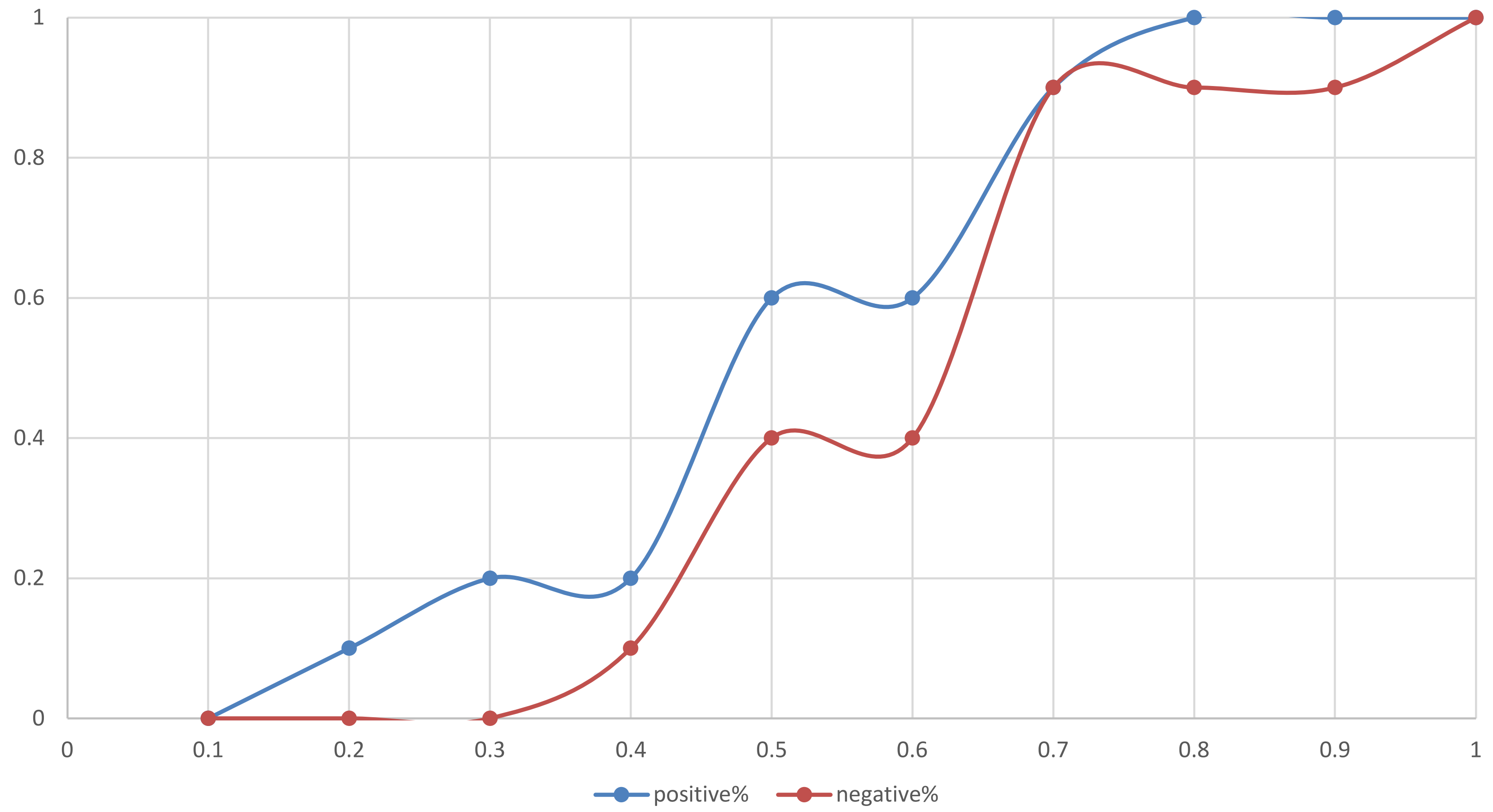
还是以上面20个样本的例子为例，将20个正负样本根据模型预测为正样本的概率，将其从高到低排序，并将其按照指定的分组进行计数，便可得：

预测概率区间	实际为正样本的样本量	实际为负样本的样本量
(0.9,1]	0	0
(0.8,0.9]	1	0
(0.7,0.8]	1	0
(0.6,0.7]	0	1
(0.5,0.6]	4	3
(0.4,0.5]	0	0
(0.3,0.4]	3	5
(0.2,0.3]	1	0
(0.1,0.2]	0	0
(0,0.1]	0	1
合计	10	10

据此我们可以每个预测概率区间中，实际正样本累积量占实际正样本总量的比率(positive%)，以及实际负样本累积量占实际负样本总量的比率(negative%)：

预测概率区间	实际为正样本的样本累计量	positive%	实际为负样本的累计样本量	negative%	|positive%-negative%|
(0.9,1]	0	0%	0	0%	0%
(0.8,0.9]	1	10%	0	0%	10%
(0.7,0.8]	2	20%	0	0%	20%
(0.6,0.7]	2	20%	1	10%	10%
(0.5,0.6]	6	60%	4	40%	20%
(0.4,0.5]	6	60%	4	40%	20%
(0.3,0.4]	9	90%	9	90%	0%
(0.2,0.3]	10	100%	9	90%	10%
(0.1,0.2]	10	100%	9	90%	10%
(0,0.1]	10	100%	10	100%	0%

由表可得，该例子的KS值为0.2，说明该例子中的模型没有区分度。将positive%与negative%绘制成图可得KS曲线图，即：



python例题

本次例题是使用波士顿房价数据集做线性回归，这是一个非常经典的数据集，**原始数据有14 个**变量的506**个**观察值,**其中**，medv(**自住房屋房价中位数**,单位:**千美 元)是原始的目**标变量，**其他**变量包括: crim (**城**镇的人均犯罪率)、zn (**占地面**积**超过25000 平方英尺的住宅用地的比例)**、indus (**每个**镇的非**零售**业务比例，单位:**英**亩)、chas (**有关**查尔斯河的虚拟变量，**如果挨着河**为1,否则为0)......等。

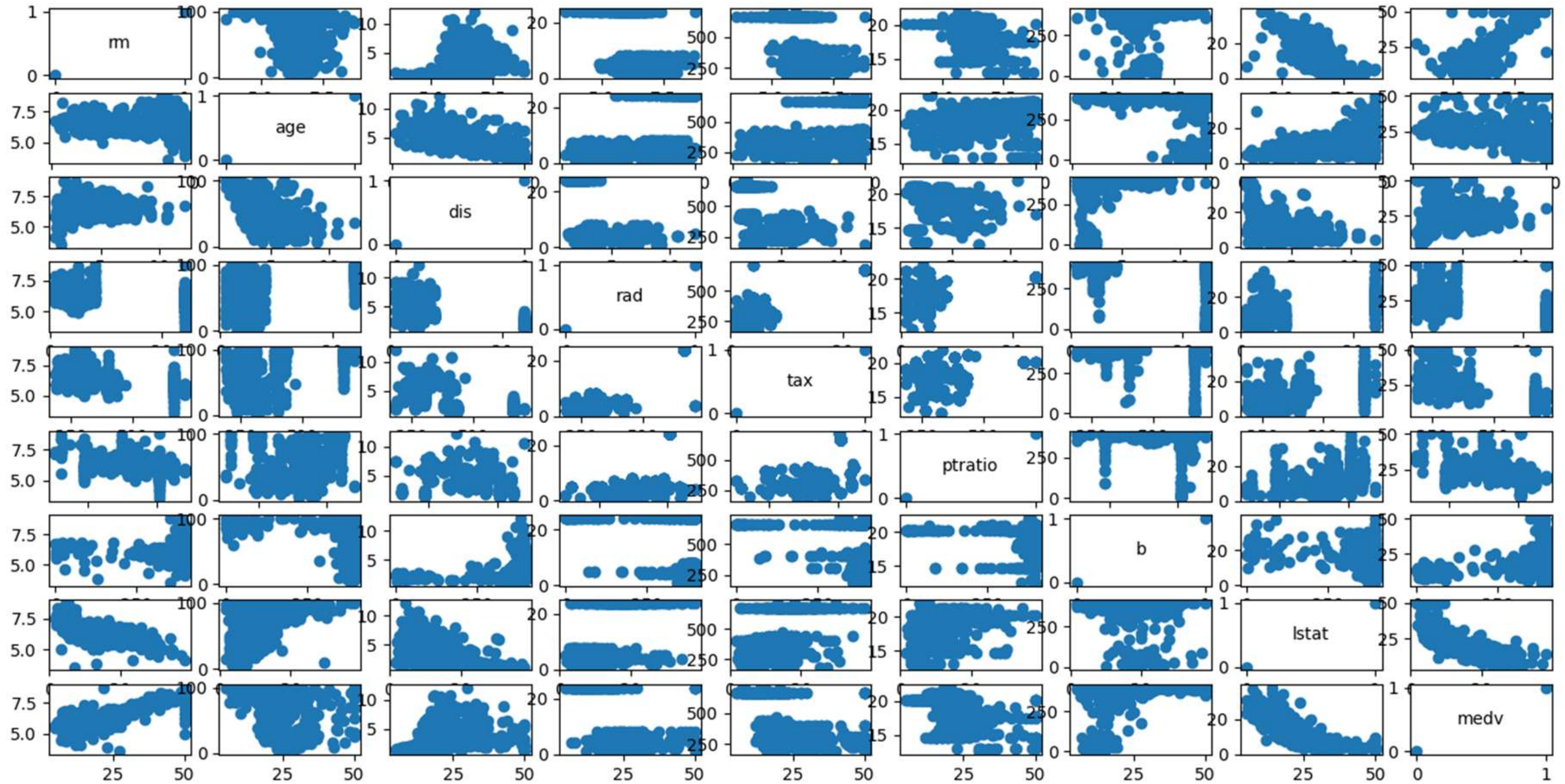
crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9

python代码：

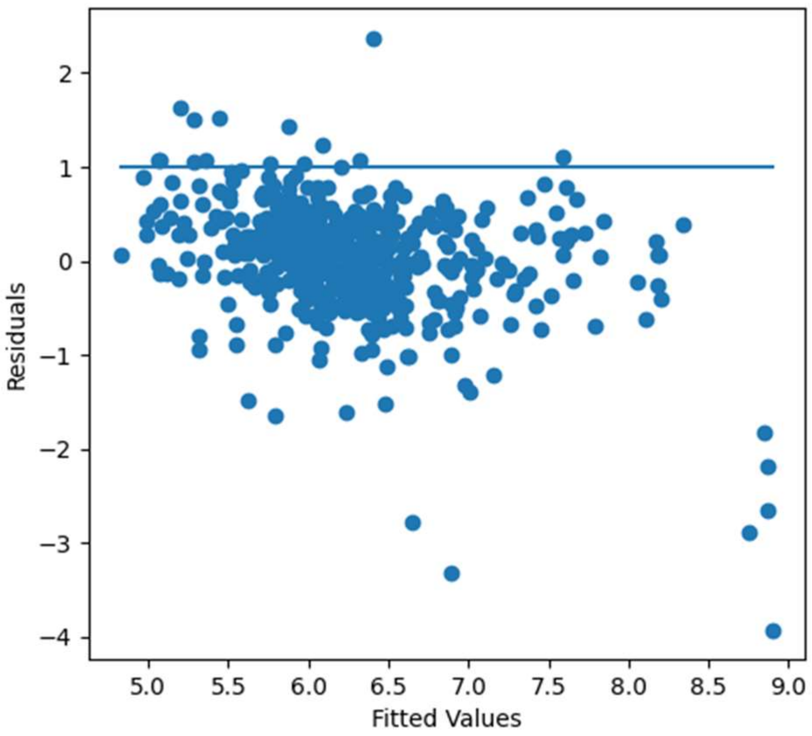
```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import scipy.stats as stats
import pandas as pd
w=pd.read_csv('BostonHousing2.csv')
wn0=w.columns
wn=wn0[5:]#不选前面不参与建模的5个变量
print(wn)
f=plt.figure(figsize=(16,8))
k=0
for i in range(len(wn)):
    for j in range(len(wn)):
        k=k+1
        if i!=j:
            f.add_subplot(len(wn),len(wn),k)
            plt.scatter(w[wn[i]],w[wn[j]])
        else:
            f.add_subplot(len(wn),len(wn),k)
            plt.scatter([0,1],[0,1])
            plt.text(.5,.5,wn[i],\
            ha='center',va='center',size=10)#打印变量名字
```

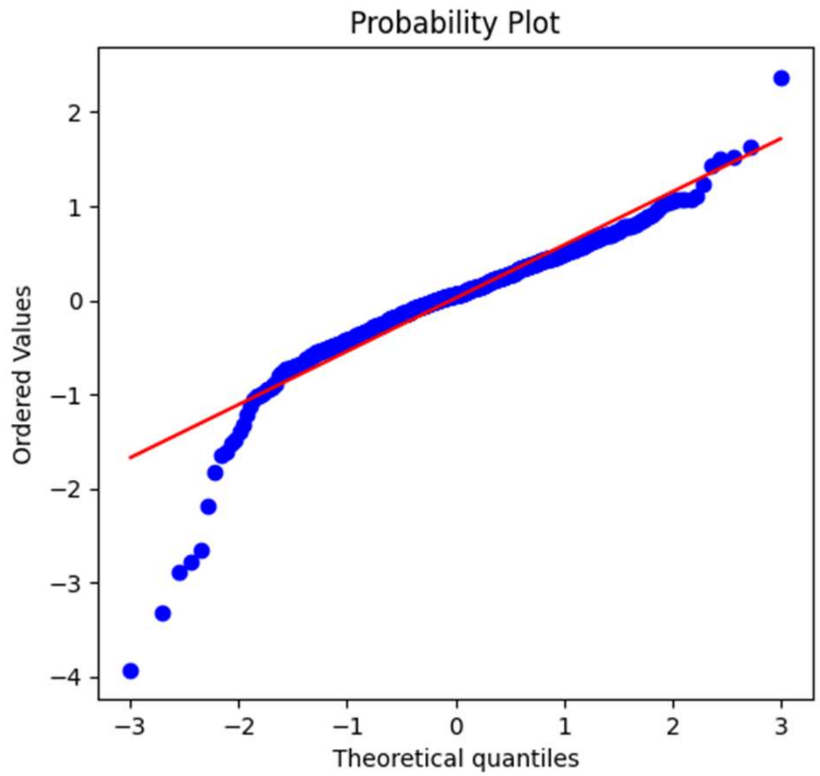
```
#确定自变量和因变量，并使用模块statsmodels.api,用最小二乘法回归拟合数据
y=np.array(w[wn[0]])[:,np.newaxis]#转成列向量
X=np.array(w[wn[1:]])
import statsmodels.api as sm
mod= sm.OLS(y,X)
res=mod.fit()
print(res.summary())
#下面用另一个模块（sklearn)做上述回归（不用截距项（，并画出残差对拟合值图和残差的Q-Q图
from sklearn import linear_model
regr =linear_model.LinearRegression(fit_intercept=False)#不做常数项
regr.fit(X,y)
print(regr.coef_)#输出估计的系数
#yhat=X.dot(regr.coef_.reshape(10,1) #直接计算拟合值
#resid =y -yhat #直接计算残差
res=y-regr.predict(X)
import pylab
res.shape=res.shape[0]#样本量
f=plt.figure(figsize=(12,5))
f.add_subplot(121)
#画残差对拟合值图
......
```

```
                            OLS Regression Results
=======================================================================================
Dep. Variable:                      y   R-squared (uncentered):                   0.991
Model:                            OLS   Adj. R-squared (uncentered):              0.991
Method:                 Least Squares   F-statistic:                              6898.
Date:                Sun, 19 Dec 2021   Prob (F-statistic):                        0.00
Time:                        11:18:24   Log-Likelihood:                         -457.85
No. Observations:                 506   AIC:                                      931.7
Df Residuals:                     498   BIC:                                      965.5
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0107      0.002      7.095      0.000       0.008       0.014
x2             0.1627      0.018      8.928      0.000       0.127       0.198
x3            -0.0205      0.007     -2.784      0.006      -0.035      -0.006
x4             0.0016      0.000      4.028      0.000       0.001       0.002
x5             0.1384      0.011     12.819      0.000       0.117       0.160
x6          -9.18e-05      0.000     -0.276      0.783      -0.001       0.001
x7             0.0007      0.006      0.114      0.909      -0.011       0.013
x8             0.0856      0.004     23.422      0.000       0.078       0.093
==============================================================================
Omnibus:                      206.030   Durbin-Watson:                   0.875
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1636.809
Skew:                          -1.568   Prob(JB):                         0.00
Kurtosis:                      11.234   Cond. No.                         402.
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[[ 1.07231413e-02  1.62662894e-01 -2.04672105e-02  1.57922115e-03
   1.38438893e-01 -9.17966904e-05  7.06264737e-04  8.56119312e-02]]

Process finished with exit code 0
```



波士顿房价数据的成对散点图





波士顿房价数据的残差对拟合值图和残差的Q-Q图