线性回归模型

上一个案例介绍的逻辑回归模型属于对数线性回归模型，而线性回归和非线性回归都同属于回归模型，接下来我们简单介绍下回归分析，一元和多元线性回归以及线性回归和逻辑回归之间的关系。

1. 回归分析

回归分析是数理统计学的一个重要组成部分，它的任务是研究变量之间的相关关系，建立变量之间的经验公式（即回归函数），以便达到预测和控制的目的。

在回归分析中：

变量 y 被称为因变量，处在被解释的地位；

变量 x 被称为自变量，用于预测因变量的变化。

按照涉及的自变量的个数，回归模型可以被分为一元回归和多元回归；

按照自变量和因变量之间的关系类型，回归模型可以被分为线性回归和非线性回归。

注：非线性回归的自变量和因变量在图像中常表现为曲线、分段等特性，暂不做介绍。

2. 一元线性回归模型

对于只涉及一个自变量的简单线性模型，可以表示为：

$$y = \beta_0 + \beta_1 x$$

式中：

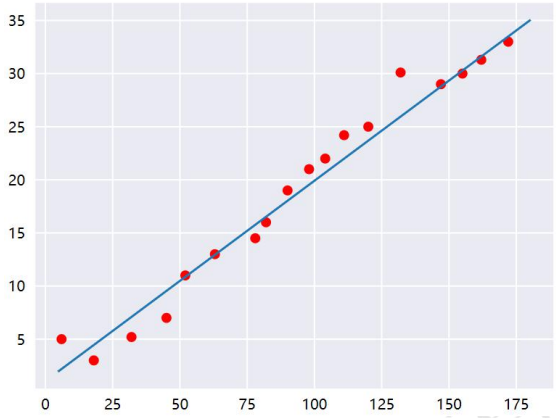
$\beta_0 + \beta_1 x$ 反映了 x 的变化而引起的 y 的线性变化，β_0 和 β_1 称为模型的参数（或权重）；

一元线性回归模型的目标就是基于已知的 n 个样本的参数（ $\hat{\beta}_0$ 和 $\hat{\beta}_1$ ）估计未知总

体的回归参数（ β_0 和 β_1 ），因此一元线性回归模型的回归方程为：

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

用图像表示回归方程如下：



直角坐标系中的横轴表示 x，纵轴表示 y，红点表示所有的样本。一元线性回归模型就是要找到一条直线，该直线能够最好的描述直角坐标系中的所有的点。这样的过程就称为**拟合**，得到的回归线的方程即为回归方程。

3. 多元线性回归模型

在许多实际问题中，影响因变量的因素往往有多个，这种一个因变量与多个自变量的回归问题就是多元回归，当因变量与各自变量之间为线性关系时，称为**多元线性回归模型**。多元线性回归分析的原理同一元线性回归基本相同。

因此涉及 p 个自变量的多元线性回归模型的回归方程为：

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p$$

因为一般的回归模型通常涉及到多个特征，因此线性回归通常是指多元线性回归模型。多元线性回归模型的目标同样是在最小化误差项的同时获得线性组合的参数(权重) 向量（ $\hat{\beta}_0$ ， $\hat{\beta}_1$ ， $\hat{\beta}_2$ ，.... ， $\hat{\beta}_p$ ）。

4. 最小二乘法

最小二乘法（method of least squares）是模型优化的方法，它是通过使因变量对应的实际 y 与直线上估计值 $\hat{y}$ 之间差的平方和达到最小时来估计 β_0 和 β_1 的方法，即：

$$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \text{ 最小时，估计权重 } \beta_0 \text{ 和 } \beta_1 \text{。}$$

该函数被称为：线性回归模型的**损失函数**。

小知识：对于图像中样本的分布，用于描述 x 和 y 关系的直线有多条，究竟用哪条直线来代表两者之间的关系，我们自然会想到距离样本点最近的一条直线，用它代表 x 和 y 之间的关系与实际数据的误差比其他任何直线都小，科学家提出了最小二乘法。

5. 正则化

在实际应用中，为了避免模型过拟合，一个可行的思路就是尽量减少模型的复杂度，即在减少误差项的同时也引入正则化项。此时，模型优化的目标就是最小化损失函数和最小化模型复杂度。

在实际的优化函数中，模型复杂度有一个可调整的系数，用于定义正则化的力度。

（1）$l1$ 正则化

模型中权重非零的特征数作为模型的复杂度，对应的计算公式为特征权重的绝对值之和，称为 $l1$ **正则化**：

$$\lambda\sum_{i=1}^{n}\left|\hat{\beta}_i\right|$$

（2）$l2$ 正则化

模型中所有特征的权重之和作为模型的复杂度，对应的计算公式为特征权重的平方和，称为 $l2$ **正则化**：

$$\lambda\sum_{i=1}^{n}\hat{\beta}_i^2$$

（3）$l1$ 正则化和 $l2$ 正则化比较

①$l1$计算的是权重的绝对值，$l2$计算的是权重的平方，因此$l1$对权重并没有$l2$那么敏感。

②使用$l1$正则化会使得特征权重更趋向于完全的零，即$l1$正则化项保证参数相对稀疏；使用$l2$正则化会使得越接近零的特征权重对模型复杂度的影响越小，越大的权重影响越明显。

③$l1$正则化得到稀疏权重矩阵，从参数相对稀疏上降低模型的复杂度；$l2$正则化从权重足够小（但不为零），使得模型的复杂度足够小。

小知识：一定程度两种正则化都可以有效防止过拟合，对于线性回归模型，加入 l2 正规化项为岭（ridge）回归；加入 l1 正规化项为套索（lasso）回归，两者都加上则为 elasticnet 回归。

6. 线性回归模型的几个统计学评估指标

（1）拟合优度系数R^2

拟合优度（Goodness of Fit）是指回归直线对观测值的拟合程度。度量拟合优度的统计量**拟合优度系数**R^2（样本决定系数、测定系数、判定系数）是一个比值结果，最大值为 1。

R^2取值接近 1，说明回归直线对观测值的拟合程度越好；反之，说明回归直线对观测值的拟合程度越差。一般来说，取值大于 0.6 则表示相对较高的线性相关性。

（2）显著性检验

推断统计学中提到假设检验一般分为两步：①做出假设②进行检验。显著性检验是回归分析中判断总体的真实情况与原假设是否有显著性差异的主要方法，包括两方面内容（统计量）：一是线性关系的检验（F 检验）；二是特征权重的检验（t 检验）。

显著性检验-F 检验：用于检验自变量和因变量之间的线性关系整体上是否显著，如果小于 0.05 则表示模型足够显著，能够揭示自变量与因变量之间存在回归关系。

F 检验与拟合优度系数 R^2 看似都是对线性关系的检验，但 F 检验仅用于检验线性关系是否显著，R^2 却重点在于表示显著情况下拟合的效果。

显著性检验-t 检验：对于每一个自变量 x_i ，用于检验自变量 x_i 对因变量 y 的影响是否显著，如果小于 0.05 则表示该项权重足够显著，该项 x_i 与 y 之间存在回归关系。

对于一般线性回归，可以使用 t 检验筛选变量，保留小于 0.05 的变量后逐步拟合回归模型；对于 lasso 模型，通过增强 l1 正则化项的稀疏力度就可以避免结果模型过拟合，但同时会丧失一些统计指标检验。

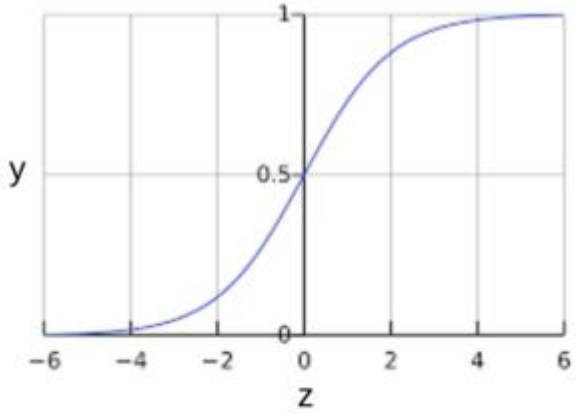
7. 线性回归与逻辑回归

在上一个课程案例中我们介绍了逻辑回归模型，逻辑回归可以说是基于线性回归而产生的模型。线性回归是用于预测因变量 y 的具体取值的模型，这一思路的前提是 y 是连续变量，但是很多情况下，需要预测的是离散变量，比如某一事件是否发生、某一样本属于哪一类等等。逻辑回归就是通过增加 sigmoid 层将这些离散变量问题转化为概率的估计问题。

在逻辑回归中，为了使得模型输出是 0-1 的一个概率，在线性回归方程的基础上，引入 sigmoid 方程：

$$\frac{1}{1+e^{-z}}$$

sigmoid 方程的图形如下图所示：



其中线性回归方程：

$$z = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p$$

注：sigmoid 方程在机器学习领域应用非常广泛，很多神经网络在输出层前都有一个 sigmoid 层。

结果分析

通过网格搜索超参数后，得到最优超参数组合并建立最终模型，通过对最终模型进行评估，得到如下图所示评估参数：

```
零特征占比 sparsity: 77.50%
R^2 square of train: 62.08%
R^2 square of test: 63.70%
```

首先稀疏度较高，模型相对并不复杂；

其次基于训练集和测试集建立起的模型的 r2_score 得分较为相近，且 test_score 略高于 train_score，说明建立起的模型没有过拟合，分析结果是比较可靠的。

而在模型训练中，35 个特征中保留特征了 9 个特征，根据特征权重大小将 9 个进行排序，如下所示：

英文名称	特征名称	权重
liability	负债总计	0.4608
asset	资产总计	0.2137
listing_days	挂牌时长	0.0421
registered_capital	注册资本	0.0353
ave_executive_age	高管平均年龄	0.0008
ave_executive_edu	高管平均学历	-0.0041
income_gr	营业收入增长率	-0.0095
asset_gr	总资产增长率	-0.0862
current_rt	流动比率	-0.0957

liability、asset 作为权重相对较高的两个影响因素，可以理解为间接融资会导致公司资产负债表中资产和负债的增加，资产总计、负债总计和融资金额具有高度相关性；而资产=流动资产+非流动资产，企业中资产具有一定的基数，所以相对负责而言，资产总计的相关性较弱些。

listing_days 挂牌时间对融资金额的影响与逻辑回归中的分析相类似。有两种可能的原因：①挂牌时间越长，公司更有可能融到更多资金；②挂牌时间较短的公司，在各

方面的准备尚不成熟，融资金额较少。

registered_capital 反应的是公司本身的素质，注册资本越多，公司起步的水平较高，对融资金额有正向影响。

ave_executive_age、ave_executive_edu 是与高管特征相关。这说明对于公司的各项运营指标来说，高管具有一定的影响力。但是高管的影响是相当复杂难以说明的，例如高管的年龄是正向相关而学历是负向相关。年龄更大间接融资金额更大是具有较为合理的解释即年龄更大，在融资领域的经验更加丰富，社交网络更加广泛。但是另一方面学历表现出负相关难以简单说明。我们此时可以引入多种假设：新三板中公司的体量虽然远比不了上市公司，但是仍然属于当地有相对稳定的公司，对比当下创业公司来讲具有更长的运营时间，所以公司高管的年龄普遍较大，学历应当考虑其当时的教学条件水平。或者因为高学历人才往往集中在一线城市，而许多公司是分布在全国各地的，它们首先与当地银行和政府的关系相对较好，但不一定能吸引高学历人才。我们不能草率地得出“学历对于创业是负向作用”这样的结论，这样的结论社会认可度不高，而且往往并不是真实情况。

asset_gr, income_gr 与间接融资金额具有一定的负相关。资产增长率和营业收入增长率的增加都反映了公司发展较好，此时公司对间接融资需求较低，反之公司对间接融资需求增强。但在公司经营状况的变动中，对间接融资的需求的相关性较为一般。

current_rt 即流动比率，是公司流动资产和流动负债总额之比。一般来说，流动比率越高，表明公司资产的变现能力越强。但是，流动比率过高，说明公司持有的现金过多，资金利用不善，可能对融资金额有一定的负面影响。

线性回归总结

1、推断统计学主要包含哪两方面内容？其中都包含哪些基本概念？

推断统计学主要包含参数估计和假设检验两方面内容：

参数估计：是指在抽样及抽样分布的基础上，根据样本统计量来推断所关心的总体参数，即利用样本数据推断总体特征参数；

假设检验：是利用样本数据判断对总体的假设是否成立。

包含以下几个概念：

（1）置信区间：由样本统计量所构造的总体参数的估计区间；

（2）置信水平：衡量总体均值落在置信区间的可能性大小；

（3）显著性水平：是指未落在置信区间的可能性大小数值；

（4）原假设和备择假设

原假设和备择假设是对总体参数或分布形式提出的两种假设。

例如今年某地新生婴儿的平均体重这一参数 μ_0，根据去年的数据 3190 克提出：

$$H_0：\mu_0=3190（g）$$

这里 H_0 表示一个事件，即今年某地新生婴儿的平均体重为 3190 克，如果该假设（H_0）不成立，则需要提出另一种假设：

$$H_1：\mu_1\neq3190（g）$$

该假设称：备择假设，即今年某地新生婴儿的平均体重不为 3190 克。

我们可以看出，原假设与备择假设互斥，肯定原假设，意味着放弃备择假设，否定原假设，意味着接受备择假设。

（5）α错误和β错误

α错误：当原假设 H_0 为真却被我们拒绝了，犯这种错误的概率用α表示，所以α错误也被称为弃真错误；

β错误：当原假设 H_0 为伪我们缺没有拒绝，犯这种错误的概率用β表示，所以β错误也被称为取伪错误。

2、线性回归和逻辑回归模型的联系是什么？其中线性回归模型涉及了哪些统计学评估指标？

（1）线性回归和逻辑回归模型的联系

逻辑回归是基于线性回归而产生的模型。线性回归是用于预测因变量 y 的具体取值的模型，这一思路的前提是 y 是连续变量，但是很多情况下，需要预测的是离散变量，比如某一事件是否发生、某一样本属于哪一类等等。逻辑回归就是通过增加 sigmoid 层将这些离散变量问题转化为概率的估计问题。

在逻辑回归中，为了使得模型输出是 0-1 的一个概率，在线性回归方程的基础上，引入 sigmoid 方程：

$$\frac{1}{1+e^{-z}}$$

其中线性回归方程：

$$z=\hat{\beta}_0+\hat{\beta}_1x_1+\hat{\beta}_2x_2+\ldots+\hat{\beta}_px_p$$

（2）线性回归模型的统计学评估指标

①拟合优度系数 R^2 ：是指回归直线对观测值的拟合程度度的统计量，取值接近 1，说明回归直线对观测值的拟合程度越好；反之，说明回归直线对观测值的拟合程度越差。

②显著性检验：是回归分析中判断总体的真实情况与原假设是否有显著性差异的主要方法，包括两方面内容（统计量）：一是线性关系的检验（F 检验）；二是特征权重的

检验（t 检验）。

3、$l1$正则化和$l2$正则化比较有哪些区别？

（1）$l1$计算的是权重的绝对值，$l2$计算的是权重的平方，因此$l1$对小权重并没有$l2$那么敏感。

（2）使用$l1$正则化会使得特征权重更趋向于完全的零，即$l1$正则化项保证参数相对稀疏；使用$l2$正则化会使得越接近零的特征权重对模型复杂度的影响越小，越大的权重影响越明显。

（3）$l1$正则化得到稀疏权重矩阵，从参数相对稀疏上降低模型的复杂度；$l2$正则化从权重足够小（但不为零），使得模型的复杂度足够小。

（4）线性回归模型中加入$l2$正规化项为岭（ridge）回归；加入$l1$正规化项为套索（lasso）回归，两者都加上则为 elasticnet 回归。

4、如何使用交叉验证分割好的数据集的进行超参数的网格搜索？

网格搜索的结果是，不同的超参数组合对应不同的评价分值，因此：

①首先需要新建 DataFrame 保存搜索结果，代码如下：

```
result_frame = pd.DataFrame(columns=['alpha', 'tol', 'validate_score', 'sparsity'])
```

这里我们需要筛选的超参数：alpha、tol

alpha：优化函数中与 l1 正则化项相乘的常数，用于调整正则化的力度。越大的值表示正则化力度更强，对变量的选择能力越强，结果将更加稀疏。（功能与逻辑回归中的参数 C 相类似）；

tol：（同逻辑回归）收敛阈值，越小则要求收敛的稳定性越高，需要的收敛时间越长。

以及记录每次使用验证集在对应模型中模型拟合优度系数 r2_score()、稀疏度。

②穷举 alpha 和 tol，例如案例中的代码：

```
for alpha in [1e-1, 7e-2, 5e-2, 3e-2, 1e-2]:
    for tol in [1e-2, 1e-3, 1e-4]:
```

以及交叉验证切分后得到训练集和验证集的索引，代码为：

```
for train_index, validate_index in kf.split(X_train, y_train):
```

③获得训练集和验证集

```
X_t, y_t = X_train[train_index], y_train[train_index]
X_v, y_v = X_train[validate_index], y_train[validate_index]
```

④创建模型，并使用交叉验证分割中得到的训练集训练模型，例如案例中的代码：

```
mod = Lasso(alpha=alpha, tol=tol, random_state=0)
mod.fit(X_t, y_t)
```

注意：收敛问题。（中级案例一逻辑回归的总结中有文字总结，可供参考）

⑤利用训练好的模型对验证集进行预测，得到模型的评估结果，代码为：

```
y_pred = mod.predict(X_v)
validate_score = r2_score(y_v, y_pred)
sparsity = np.mean(mod.coef_ == 0)
```

⑥最后就是将参数搜索结果放到新建 DataFrame 中，例如案例中的代码：

```
result_frame=result_frame.append({'alpha':alpha,'tol':tol,'validate_score':validate_score,'sparsity':sparsity}, ignore_index=True)
```

⑦打印结果，并拟定最优超参数组合

在整个过程中，需要注意的是代码书写的格式，以及 for 语句和 if 语句后的冒号。

5、如何查看最终模型是否存在过拟合的情况？

基于训练好的最终模型，通过对比训练集和测试集分别在模型中的表现来查看最终模型是否存在过拟合的情况，这里需要注意三点：

（1）使用交叉验证得到训练集和验证集只是为了搜索超参数组合，其中使用训练集得到的模型不能作为最终模型，最终模型仍然需要使用整个训练集进行训练得到；

（2）使用训练集训练得到的模型，再次预测训练集得到的结果，并非与原标记完全一致。

（3）关于模型的过拟合问题，除了对比模型在训练集和测试集的拟合优度系数大小关系外，稀疏度可以反映模型的复杂程度，也可以作为模型过拟合的一个判断因素。

该部分内容及输出结果代码如下所示：

```
# 在最优参数组合下重新创建套索回归模型
mod = Lasso(alpha=0.05, tol=0.01, random_state=0)
# 使用训练集训练模型
mod.fit(X_train, y_train)

# 使用训练好的模型预测
y_train_pred = mod.predict(X_train)
y_test_pred = mod.predict(X_test)

# 评估模型
# 计算训练集和测试集的R^2
train_score = r2_score(y_train, y_train_pred)
test_score = r2_score(y_test, y_test_pred)
# 计算模型稀疏度
sparsity = np.mean(mod.coef_ == 0)

# 结果输出
print("零特征占比 Sparsity: %.2f%%" % (sparsity * 100))
print("R^2 square of train: %.2f%%" % (train_score * 100))
print("R^2 square of test: %.2f%%\n" % (test_score * 100))

print("以下为非零特征及对应系数：")
print("%20s\tcoef" % "feature_name")
for name, coef in zip(x_mlapper.transformed_names_, mod.coef_):
    if coef == 0:
        continue
    print("%20s\t%.4f" % (name, coef))
```