超级干货 ：一文读懂回归分析

前言

1.“回归”一词的由来

我们不必在“回归”一词上费太多脑筋。英国著名统计学家弗朗西斯·高尔顿（Francis Galton,1822—1911）是最先应用统计方法研究两个变量之间关系问题的人。“回归”一词就是由他引入的。他对父母身高与儿女身高之间的关系很感兴趣，并致力于此方面的研究。高尔顿发现，虽然有一个趋势：父母高，儿女也高；父母矮，儿女也矮，但从平均意义上说，给定父母的身高，儿女的身高却趋同于或者说回归于总人口的平均身高。换句话说，尽管父母双亲都异常高或异常矮，儿女身高并非也普遍地异常高或异常矮，而是具有回归于人口总平均高的趋势。更直观地解释，父辈高的群体，儿辈的平均身高低于父辈的身高；父辈矮的群体，儿辈的平均身高高于其父辈的身高。用高尔顿的话说，儿辈身高的“回归”到中等身高。这就是回归一词的最初由来。

回归一词的现代解释是非常简洁的：回归时研究因变量对自变量的依赖关系的一种统计分析方法，目的是通过自变量的给定值来估计或预测因变量的均值。它可用于预测、时间序列建模以及发现各种变量之间的因果关系。

使用回归分析的益处良多，具体如下：
1) 指示自变量和因变量之间的显著关系；
2) 指示多个自变量对一个因变量的影响强度。
回归分析还可以用于比较那些通过不同计量测得的变量之间的相互影响，如价格变动与促销活动数量之间的联系。这些益处有利于市场研究人员，数据分析人员以及数据科学家排除和衡量出一组最佳的变量，用以构建预测模型。

2.为什么使用回归分析

1）更好地了解

对某一现象建模，以更好地了解该现象并有可能基于对该现象的了解来影响政策的制定以及决定采取何种相应措施。基本目标是测量一个或多个变量的变化对另一变量变化的影响程度。示例：了解某些特定濒危鸟类的主要栖息地特征（例如：降水、食物源、植被、天敌），以协助通过立法来保护该物种。

2）建模预测

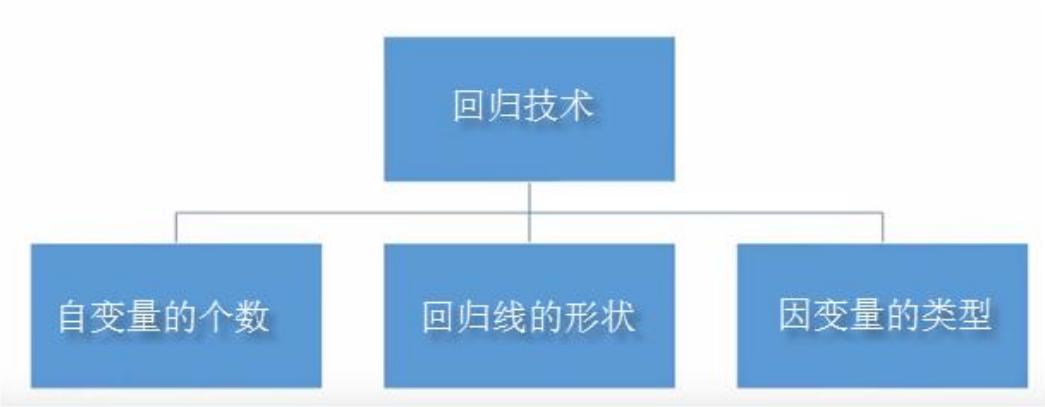
对某种现象建模以预测其他地点或其他时间的数值。基本目标是构建一个持续、准确的预测模型。示例：如果已知人口增长情况和典型的天气状况，那么明年的用电量将会是多少？

3）探索检验假设

还可以使用回归分析来深入探索某些假设情况。假设您正在对住宅区的犯罪活动进行建模，以更好地了解犯罪活动并希望实施可能阻止犯罪活动的策略。开始分析时，您很可能有很多问题或想要检验的假设情况。
回归分析的作用主要有以下几点：
1）挑选与因变量相关的自变量；
2）描述因变量与自变量之间的关系强度；
3）生成模型，通过自变量来预测因变量；
4）根据模型，通过因变量，来控制自变量。

回归分析方法

现在有各种各样的回归技术可用于预测，这些技术主要包含三个度量：自变量的个数、因变量的类型以及回归线的形状。



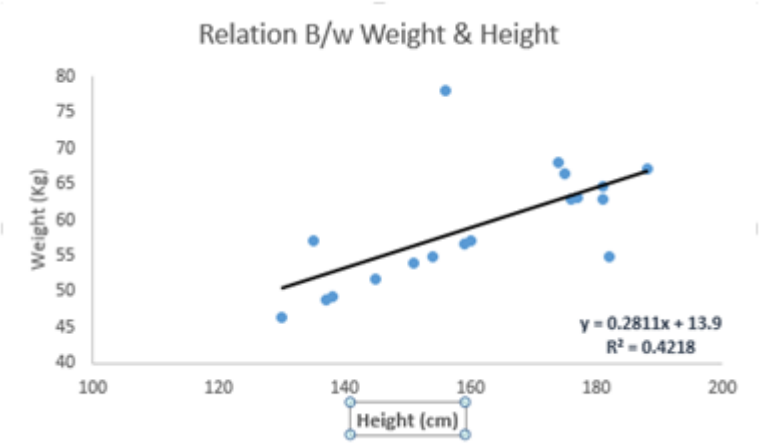
1.回归分析方法

1）线性回归

线性回归它是最为人熟知的建模技术之一。线性回归通常是人们在学习预测模型时首选的少数几种技术之一。在该技术中，因变量是连续的，自变量（单个或多个）可以是连续的也可以是离散的，回归线的性质是线性的。线性回归使用最佳的拟合直线（也就是回归线）建立因变量 (Y) 和一个或多个自变量 (X) 之间的联系。用一个等式来表示它，即：

Y=a+b*X + e

其中 a 表示截距，b 表示直线的倾斜率，e 是误差项。这个等式可以根据给定的单个或多个预测变量来预测目标变量的值。



一元线性回归和多元线性回归的区别在于，多元线性回归有一个以上的自变量，而一元线性回归通常只有一个自变量。

线性回归要点：

1）自变量与因变量之间必须有线性关系；

2）多元回归存在多重共线性，自相关性和异方差性；

3）线性回归对异常值非常敏感。它会严重影响回归线，最终影响预测值；

4） 多重共线性会增加系数估计值的方差，使得估计值对于模型的轻微变化异常敏感，结果就是系数估计值不稳定；

5）在存在多个自变量的情况下，我们可以使用向前选择法，向后剔除法和逐步筛选法来选择最重要的自变量。

2）Logistic 回归

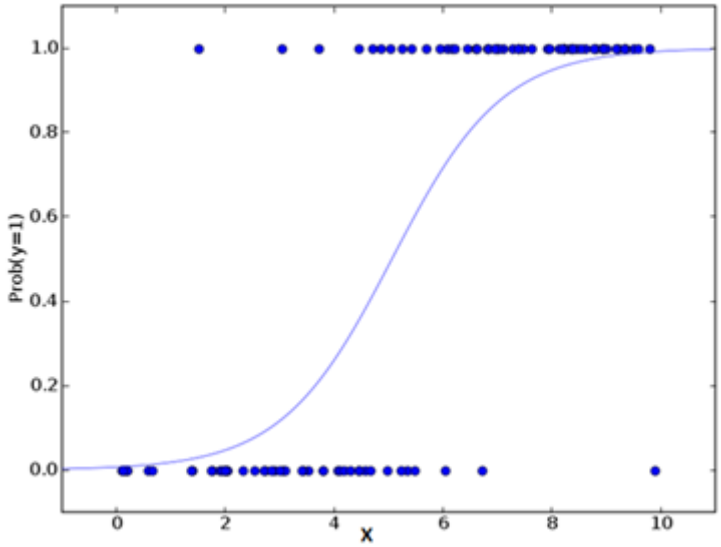
Logistic 回归可用于发现 “事件=成功”和“事件=失败”的概率。当因变量的类型属于二元（1 / 0、真/假、是/否）变量时，我们就应该使用逻辑回归。这里，Y 的取值范围是从 0 到 1，它可以用下面的等式表示：

odds= p/ (1-p) = 某事件发生的概率/ 某事件不发生的概率

ln(odds) = ln(p/(1-p))

logit(p) = ln(p/(1-p)) =b0+b1X1+b2X2+b3X3....+bkXk

如上，p 表述具有某个特征的概率。在这里我们使用的是的二项分布（因变量），我们需要选择一个最适用于这种分布的连结函数。它就是 Logit 函数。在上述等式中，通过观测样本的极大似然估计值来选择参数，而不是最小化平方和误差（如在普通回归使用的）。



Logistic 要点：
1）Logistic 回归广泛用于分类问题；
2）Logistic 回归不要求自变量和因变量存在线性关系。它可以处理多种类型的关系，因为它对预测的相对风险指数使用了一个非线性的 log 转换；
3）为了避免过拟合和欠拟合，我们应该包括所有重要的变量。有一个很好的方法来确保这种情况，就是使用逐步筛选方法来估计 Logistic 回归；
4）Logistic 回归需要较大的样本量，因为在样本数量较少的情况下，极大似然估计的效果比普通的最小二乘法差；
5） 自变量之间应该互不相关，即不存在多重共线性。然而，在分析和建模中，我们可以选择包含分类变量相互作用的影响；
6）如果因变量的值是定序变量，则称它为序 Logistic 回归；
7）如果因变量是多类的话，则称它为多元 Logistic 回归。

3）Cox 回归

Cox 回归的因变量就有些特殊，它不经考虑结果而且考虑结果出现时间的回归模型。它用一个或多个自变量预测一个事件（死亡、失败或旧病复发）发生的时间。Cox 回归的主要作用发现风险因素并用于探讨风险因素的强弱。但它的因变量必须同时有 2 个，一个代表状态，必须是分类变量，一个代表时间，应该是连续变量。只有同时具有这两个变量，才能用 Cox 回归分析。Cox 回归主要用于生存资料的分析，生存资料至少有两个结局变量，一是死亡状态，是活着还是死亡；二是死亡时间，如果死亡，什么时间死亡？如果活着，从开始观察到结束时有多久了？所以有了这两个变量，就可以考虑用 Cox 回归分析。

4）poisson 回归

通常，如果能用 Logistic 回归，通常也可以用 poission 回归，poisson 回归的因变量是个数，也就是观察一段时间后，发病了多少人或是死亡了多少人等等。其实跟 Logistic 回归差不多，因为 logistic 回归的结局是是否发病，是否死亡，也需要用到发病例数、死亡例数。

5）Probit 回归

Probit 回归意思是“概率回归”。用于因变量为分类变量数据的统计分析，与 Logistic 回归近似。也存在因变量为二分、多分与有序的情况。目前最常用的为二分。医学研究中常见的半数致死剂量、半数有效浓度等剂量反应关系的统计指标，现在标准做法就是调用 Pribit 过程进行统计分析。

6）负二项回归

所谓负二项指的是一种分布，其实跟 poission 回归、logistic 回归有点类似，poission 回归用于服从 poission 分布的资料，logistic 回归用于服从二项分布的资料，负二项回归用于服从负二项分布的资料。如果简单点理解，二项分布可以认为就是二分类数据，poission 分布就可以认为是计数资料，也就是个数，而不是像身高等可能有小数点，个数是不可能有小数点的。负二项分布，也是个数，只不过比 poission 分布更苛刻，如果结局是个数，而且结局可能具有聚集性，那可能就是负二项分布。简单举例，如果调查流感的影响因素，结局当然是流感的例数，如果调查的人有的在同一个家庭里，由于流感具有传染性，那么同一个家里如果一个人得流感，那其他人可能也被传染，因此也得了流感，那这就是具有聚集性，这样的数据尽管结果是个数，但由于具有聚集性，因此用 poission 回归不一定合适，就可以考虑用负二项回归。

7）weibull 回归

中文有时音译为威布尔回归。关于生存资料的分析常用的是 cox 回归，这种回归几乎统治了整个生存分析。但其实夹缝中还有几个方法在顽强生存着，而且其实很有生命力。weibull 回归就是其中之一。cox 回归受欢迎的原因是它简单，用的时候不用考虑条件（除了等比例条件之外），大多数生存数据都可以用。而 weibull 回归则有条件限制，用的时候数据必须符合 weibull 分布。如果数据符合 weibull 分布，那么直接套用 weibull 回归自然是最理想的选择，它可以给出最合理的估计。如果数据不符合 weibull 分布，那如果还用 weibull 回归，那就套用错误，结果也就会缺乏可信度。weibull 回归就像是量体裁衣，把体形看做数据，衣服看做模型，weibull 回归就是根据某人实际的体形做衣服，做出来的也就合身，对其他人就不一定合身了。cox 回归，就像是到商场去买衣服，衣服对很多人都合适，但是对每个人都不是正合适，只能说是大致合适。至于到底是选择麻烦的方式量体裁衣，还是选择简单到商场直接去买现成的，那就根据个人倾向，也根据具体对自己体形的了解程度，如果非常熟悉，自然选择量体裁衣更合适。如果不大了解，那就直接去商场买大众化衣服相对更方便些。

8）主成分回归

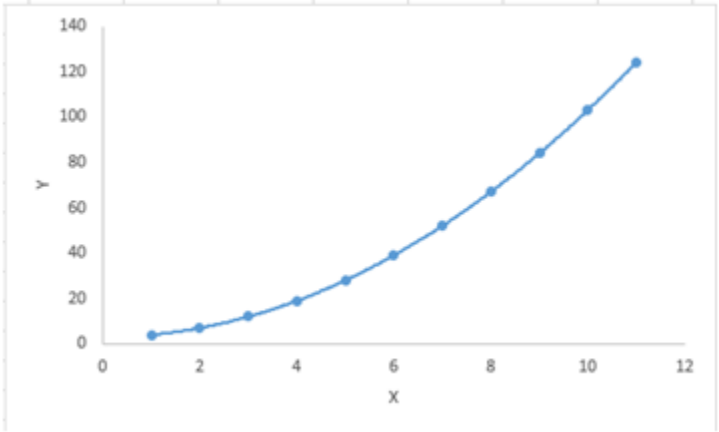
主成分回归是一种合成的方法，相当于主成分分析与线性回归的合成。主要用于解决自变量之间存在高度相关的情况。这在现实中不算少见。比如要分析的自变量中同时有血压值和血糖值，这两个指标可能有一定的相关性，如果同时放入模型，会影响模型的稳定，有时也会造成严重后果，比如结果跟实际严重不符。当然解决方法很多，最简单的就是剔除掉其中一个，但如果实在舍不得，觉得删了太可惜，那就可以考虑用主成分回归，相当于把这两个变量所包含的信息用一个变量来表示，这个变量我们称它叫主成分，所以就叫主成分回归。当然，用一个变量代替两个变量，肯定不可能完全包含他们的信息，能包含 80%或 90%就不错了。但有时候我们必须做出抉择，你是要 100%的信息，但是变量非常多的模型？还是要 90%的信息，但是只有 1 个或 2 个变量的模型？打个比方，你要诊断感冒，是不是必须把所有跟感冒有关的症状以及检查结果都做完？还是简单根据几个症状就大致判断呢？我想根据几个症状大致能确定 90%是感冒了，不用非得 100%的信息不是吗？模型也是一样，模型是用于实际的，不是空中楼阁。既然要用于实际，那就要做到简单。对于一种疾病，如果 30 个指标能够 100%确诊，而 3 个指标可以诊断 80%，我想大家会选择 3 个指标的模型。这就是主成分回归存在的基础，用几个简单的变量把多个指标的信息综合一下，这样几个简单的主成分可能就包含了原来很多自变量的大部分信息。这就是主成分回归的原理。

9）岭回归

当数据之间存在多重共线性（自变量高度相关）时，就需要使用岭回归分析。在存在多重共线性时，尽管最小二乘法（OLS）测得的估计值不存在偏差，它们的方差也会很大，从而使得观测值与真实值相差甚远。岭回归通过给回归估计值添加一个偏差值，来降低标准误差。
上面，我们看到了线性回归等式：
y=a+ b*x
这个等式也有一个误差项。完整的等式是：
y=a+b*x+e (误差项), [误差项是用以纠正观测值与预测值之间预测误差的值]
=> y=a+y= a+ b1x1+ b2x2+....+e, 针对包含多个自变量的情形。
在线性等式中，预测误差可以划分为 2 个分量，一个是偏差造成的，一个是方差造成的。预测误差可能会由这两者或两者中的任何一个造成。在这里，我们将讨论由方差所造成的误差。岭回归通过收缩参数 λ（lambda）解决多重共线性问题。请看下面的等式：

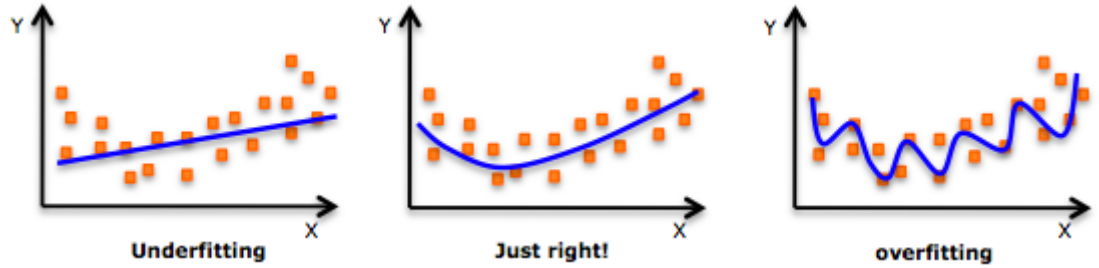
$$= \underset{\beta \in \mathbf{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

在这个等式中，有两个组成部分。第一个是最小二乘项，另一个是 β2（β-平方）和的 λ 倍，其中 β 是相关系数。λ 被添加到最小二乘项中用以缩小参数值，从而降低方差值。
岭回归要点：
1）除常数项以外，岭回归的假设与最小二乘回归相同；
2） 它收缩了相关系数的值，但没有达到零，这表明它不具有特征选择功能；
3）这是一个正则化方法，并且使用的是 L2 正则化。
10）偏最小二乘回归
偏最小二乘回归也可以用于解决自变量之间高度相关的问题。但比主成分回归和岭回归更好的一个优点是，偏最小二乘回归可以用于例数很少的情形，甚至例数比自变量个数还少的情形。所以，如果自变量之间高度相关、例数又特别少、而自变量又很多，那就用偏最小二乘回归就可以了。它的原理其实跟主成分回归有点像，也是提取自变量的部分信息，损失一定的精度，但保证模型更符合实际。因此这种方法不是直接用因变量和自变量分析，而是用反映因变量和自变量部分信息的新的综合变量来分析，所以它不需要例数一定比自变量多。偏最小二乘回归还有一个很大的优点，那就是可以用于多个因变量的情形，普通的线性回归都是只有一个因变量，而偏最小二乘回归可用于多个因变量和多个自变量之间的分析。因为它的原理就是同时提取多个因变量和多个自变量的信息重新组成新的变量重新分析，所以多个因变量对它来说无所谓。
11）多项式回归
对于一个回归等式，如果自变量的指数大于 1，那么它就是多项式回归等式。如下等式所示：
y=a+b*x^2
在这种回归技术中，最佳拟合线不是直线。而是一个用于拟合数据点的曲线。



多项式回归要点：

1）虽然存在通过高次多项式得到较低的错误的趋势，但这可能会导致过拟合。需要经常画出关系图来查看拟合情况，并确保拟合曲线正确体现了问题的本质。下面是一个图例，可以帮助理解：



2）须特别注意尾部的曲线，看看这些形状和趋势是否合理。更高次的多项式最终可能产生怪异的推断结果。

12）逐步回归

该回归方法可用于在处理存在多个自变量的情形。在该技术中，自变量的选取需要借助自动处理程序，无须人为干预。通过观察统计的值，如 R-square、t-stats 和 AIC 指标，来识别重要的变量，可以实现这一需求。逐步回归通过同时添加/去除基于指定标准的协变量来拟合模型。下面列出了一些最常用的逐步回归方法：

1）标准逐步回归法需要做两件事情，即根据需要为每个步骤添加和删除预测因子；

2）向前选择法从模型中最重要的预测因子开始，然后为每一步添加变量；

3）向后剔除法从模型中所有的预测因子开始，然后在每一步删除重要性最低的变量。

这种建模技术的目的是使用最少的预测因子变量来最大化预测能力。这也是处理高维数据集的方法之一。

13）套索回归

与岭回归类似，套索也会对回归系数的绝对值添加一个罚值。此外，它能降低偏差并提高线性回归模型的精度。看看下面的等式：

$$= \underset{\beta \in R^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

套索回归与岭回归有一点不同，它在惩罚部分使用的是绝对值，而不是平方值。这导致惩罚（即用以约束估计的绝对值之和）值使一些参数估计结果等于零。使用的惩罚值越大，估计值会越趋近于零。这将导致我们要从给定的 n 个变量之外选择变量。

套索回归要点：

1）除常数项以外，这种回归的假设与最小二乘回归类似；

2）它将收缩系数缩减至零（等于零），这确实有助于特征选择；

3）这是一个正则化方法，使用的是 L1 正则化；

4）如果一组预测因子是高度相关的，套索回归会选出其中一个因子并且将其它因子收缩为零。

14）ElasticNet 回归

ElasticNet 回归是套索回归和岭回归的组合体。它会事先使用 L1 和 L2 作为正则化矩阵进行训练。当存在多个相关的特征时，Elastic-net 会很有用。岭回归一般会随机选择其中一个特征，而 Elastic-net 则会选择其中的两个。同时包含岭回归和套索回归的一个切实的优点是，ElasticNet 回归可以在循环状态下继承岭回归的一些稳定性。

ElasticNet 回归要点：

1）在高度相关变量的情况下，它会产生群体效应；

2）选择变量的数目没有限制；

3）它可以承受双重收缩。

2.如何选择回归模型

当只了解一两种回归技术的时候，情况往往会比较简单。然而，当我们在应对问题时可供选择的方法越多，选择正确的那一个就越难。类似的情况下也发生在回归模型中。

掌握多种回归模型时，基于自变量和因变量的类型、数据的维数以及数据的其它基本特征去选择最合适的技术非常重要。以下是要选择正确的回归模型时需要考虑的主要因素：

1）数据探索是构建预测模型的不可或缺的部分。在选择合适的模型前，比如识别变量的关系和影响，应该首先执行这一步骤。

2）比较不同模型的拟合优点，我们可以分析不同的指标参数，如统计意义的参数，R-square，调整 R-square，AIC，BIC 以及误差项，另一个是 Mallows' Cp 准则。这个主要是通过将所选的模型与所有可能的子模型（或仔细挑选的一组模型）进行对比，检查可能出现的偏差。

3）交叉验证是评估预测模型最好的方法。使用该方法，需将数据集分成两份（一份用于训练，一份用于验证）。使用观测值和预测值之间的均方差即可快速衡量预测精度。

4）如果数据集中存在是多个混合变量，那就不应选择自动模型选择方法，因为我们并不愿意将所有变量同时放在同一个模型中。

5）所选择的回归技术也取决于你的目的。可能会出现这样的情况，一个不太强大的模型与具有高度统计学意义的模型相比，更易于实现。

6） 回归正则化方法（套索，岭和 ElasticNet）在高维数据和数据集变量之间存在多重共线性的情况下运行良好。

诊断回归分析结果

为了理解、解释、预测某个问题，我们会进行回归分析。但事实上，选择一组优质的自变量并不是那么容易。通常我们会根据一些常识、理论基础、某些研究、专家的意见、参考文献等等选择一组自变量，来进行自变量的筛选。因此，我们需要诊断回归分析的质量——回归分析的结果诊断。

1.自变量与因变量是否具有预期的关系

每个自变量都会有一个系数，系数具有+/-号，来表示自变量与因变量的关系。从工具的得到的报告中，我们看到的系数的正负，每个自变量应该是我们期望的关系。如果有非常不符合逻辑的系数，我们就应该考虑剔除它了。

当然，有时也可能得到与常识不同的结论。举个例子，假如我们在研究森林火灾，我们通常认为降雨充沛的区域火灾的发生率会相对较低，也就是所谓的负相关，但是，这片森林火灾频发的原因可能是闪电雷击，这样降雨量这个自变量可能就不是常识中的负相关的关系了。

因此，我们除了验证自变量的系数与先验知识是否相符外，还有继续结合其他项检查继续诊断，从而得出更可靠的结论。

2.自变量对模型是否有帮助

自变量对模型有无帮助说的就是自变量是否有显著性。那如何了解这些自变量是否有显著性呢？

如果自变量的系数为零（或非常接近零），我们认为这个自变量对模型没有帮助，统计检验就用来计算系数为零的概率。如果统计检验返回一个小概率值（p 值），则表示系数为零的概率很小。如果概率小于 0.05，汇总报告上概率（Probability） 旁边的一个 星号（*） 表示相关自变量对模型非常重要。换句话说，其系数在 95%置信度上具有统计显著性。

利用空间数据在研究区域内建模的关系存在差异是非常常见的，这些关系的特征就是不稳定。我们就需要通过 稳健概率（robust probability） 了解一个自变量是否具有统计显著性。

3.残差是否有空间聚类

残差在空间上应该是随机分布的，而不应该出现聚类。这项检查我们可以使用 空间自相关工具（Spatial Autocorrelation Tool）工具进行检查。

4.模型是否出现了倾向性

我们常说，不要戴着“有色眼镜”看人。同样，回归分析模型中，也不要带有“成见”，不能具有倾向性，否则，这不是个客观合理的模型。

我们都知道正态分布是个极好的分布模式，如果我们正确的构建了回归分析模型，那么模型的残差会符合完美的正态分布，其图形为钟形曲线。

当模型出现偏差时，可能我们看到的图形也是诡异的，这样我们就无法完全信任所预测的结果。

5.自变量中是否存在冗余

在我们建模的过程中，应尽量去选择表示各个不同方面的自变量，也就是尽量避免传达相同或相似信息的自变量。要清楚，引入了冗余变量的模型是不足以信任的。

6.评估模型的性能

最后需要做的是，评估模型的性能。 矫 R2 值是评估自变量对因变量建模的重要度量。

这项检查应该放到最后。一旦我们通过了前面的所有检验，接下来就可以进行评估矫正 R2 值。

R2 值的范围介于 0 和 1 之间，以百分比形式表示。假设正在为犯罪率建模，并找到一个通过之前所有五项检查的模型，其校正 R2 值为 0.65。这样就可以了解到模型中的自变量说明犯罪率是 65%。在有些科学领域，能够解释复杂现象的 23% 就会让人兴奋不已。在其他领域，一个 R2 值可能需要更靠近 80%或 90%才能引起别人的注意。不管采用哪一种方式，校正 R2 值都会帮我们判断自己模型的性能。

另一项辅助评估模型性能的重要诊断是修正的 Akaike 信息准则/Akaike’sinformation criterion (AIC)。AIC 值是用于比较多个模型的一项有用度量。例如，可能希望尝试用几组不同的自变量为学生的分数建模。在一个模型中仅使用人口统计变量，而在另一个模型选择有关学校和教室的变量，如每位

学生的支出和师生比。只要所有进行比较的模型的因变量（在本示例中为学生测试分数）相同，我们就可以使用来自每个模型的 AIC 值确定哪一个的表现更好。模型的 AIC 值越小，越适合观测的数据。

回归设计常用软件

目前，用于回归设计的统计软件较多，无论是对回归方案设计，还是对试验数据处理和回归设计成果的应用分析，都有相应的软件支撑，或是自编自用的专业软件，或是具有商业性质的统计软件包，多种多样，各有特色。为了便于回归设计的更好应用，这里简要地介绍挑选或评价统计软件的基本思考以及几种回归设计常用的统计软件，以利相关人员简捷地选用。

1.统计软件的选用原则

在挑选或评价统计软件时，应从以下几个方面加以考虑：

1）可用性

一个软件如果能为用户提供良好的用户界面、灵活的处理方式和简明的语句或命令，就称这个软件可用性强。随着统计软件在可用性方面的不断进步，很多统计软件的语法规则简明、灵活、学用方便，这是人们非常欢迎的。

2）数据管理

数据录人、核查、修改、转换和选择，统称为数据管理。好的软件，如 SAS(statistical analysis system)，SPSS(statistical package for thesocial science) 等的数据管理功能已近似大众化的数据库软件。统计软件与数据库软件之间建立接口，使数据管理不断深人，用起来非常方便。

3）文件管理

数据文件、程序文件、结果文件等一些文件的建立、存取、修改、合并等，统称为文件管理。它的功能越强，操作就越简单，越方便。由于操作系统本身文件管理功能较强。因此，从统计软件直接调用操作系统的命令可大大增强其文件管理功能。现在好的统计软件已设计了这类调用指令。

4）统计分析

统计分析是统计软件的核心。统计分析方法的计算机程序的数量和种类决定了数据处理的深度。有些软件，如 SAS，BMDP(biomedical computer programs)等。所包括的分析过程，足够科研与管理之需。由于统计量的选择，参数估计的方法等是多种多样的，用户往往希望统计分析过程尽可能多地提供选项，这样可以提高统计分析的灵活性和深度。

5）容量

尽管处理的数据量与计算机硬件有直接关系，然而，软件的设计和程序编写技巧仍起很大作用。软件好，在一定程度上可以弥补硬件的不足，而低水平的软件会浪费很好的硬件配置。通常，统计软件应至少能同时进行不小于 10 个变量的上千个数据点的分析、综合、对比与预测。

2.SAS 软件系统

SAS 软件系统于 20 世纪 70 年代由美国 SAS 研究所开发。SAS 软件是用于决策支援的大型集成资讯系统，但该软件系统最早的功能限于统计分析；至今，统计分析功能也仍是它的重要模组和核心功能。SAS 已经遍布全世界，重要应用领域涵盖政府的经济决策与企业的决策支援应用等，使用的单位遍及金融、医药卫生、生产、运输、通讯、科学研究、政府和教育等领域；在资料处理和统计分析领域，SAS 系统被誉统计软件界的巨无霸。SAS 是一个模块化、集成化的大型应用软件系统。它由数十个专用模块构成，功能包括数据访问、数据储存及管理、应用开发、图形处理、数据分析、报告编制、运筹学方法、计量经济学与预测等等。 SAS 系统基本上可以分为四大部分：SAS 数据库部分；SAS 分析核心；SAS 开发呈现工具；SAS 对分布处理模式的支持及其数据仓库设计。 SAS 系统主要完成以数据为中心的四大任务：数据访问；数据管理；数据呈现；数据分析。SAS 是由大型机系统发展而来，其核心操作方式就是程序驱动，经过多年的发展，现在已成为一套完整的计算机语言，其用户界面也充分体现了这一特点：它采用 MDI （多文档界面），用户在 PGM 视窗中输入程序，分析结果以文本的形式在 OUTPUT 视窗中输出。使用程序方式，用户可以完成所有需要做的工作，包括统计分析、预测、建模和模拟抽样等。但是，这使得初学者在使用 SAS 时必须要学习 SAS 语言，入门比较困难。

3.Excel 软件

在回归设计的实践中，一些计算机软件可以解决多元回归分析的求解问题，但常常是数据的输入和软件的操作运用要经过专门训练。Excel 软件为回归分析的求解给出了非常方便的操作过程，而且目前 Excel 软件几乎在每台计算机上都已经安装。

Excel 是一个面向商业、科学和工程计算的数据分析软件，它的主要优点是具有对数据进行分析、计算、汇总的强大功能。除了众多的函数功能外，Excel 的高级数据分析工具则给出了更为深入、更为有用、针对性更强的各类经营和科研分析功能。高级数据分析工具集中了 Excel 最精华、对数据分析最有用的部分，其分析工具集中在 Excel 主菜单中的“工具”子菜单内，回归分析便为其中之一。

Excel 是以电子表格的方式来管理数据的，所有的输入、存取、提取、处理、统计、模型计算和图形分析都是围绕电子表格来进行的。

4.Statistica 软件

Statistica 是由统计软件公司（Statsoft）开发、专用于科技及工业统计的大型软件包。它除了具有常规的统计分析功能外，还包括有因素分析、质量控制、过程分析、回归设计等模块。利用其回归设计模块可以进行回归正交设计、正交旋转组合设计、正交多项式回归设计、A 最优及 D 最优设计等。该软件包还可以进行对试验结果的统计检验、误差分析、试验水平估计和各类统计图表、曲线、曲面的分析计算工作。

5.SPSS 软件

SPSS 是世界上最早采用图形菜单驱动界面的统计软件，它最突出的特点就是操作界面极为友好，输出结果美观漂亮。它将几乎所有的功能都以统一、规范的界面展现出来，使用 Windows 的窗口方式展示各种管理和分析数据方法的功能，对话框展示出各种功能选择项。用户只要掌握一定的 Windows 操作技能，精通统计分析原理，就可以使用该软件为特定的科研工作服务。SPSS 采用类似 EXCEL 表格的方式输入与管理数据，数据接口较为通用，能方便的从其他数据库中读入数据。其统计过程包括了常用的、较为成熟的统计过程，完全可以满足非统计专业人士的工作需要。输出结果十分美观，存储时则是专用的 SPO 格式，可以转存为 HTML 格式和文本格式。对于熟悉老版本编程运行方式的用户，SPSS 还特别设计了语法生成窗口，用户只需在菜单中选好各个选项，然后按"粘贴"按钮就可以自动生成标准的 SPSS 程序。极大的方便了中、高级用户。

6.R 软件

R 语言是统计领域广泛使用的，诞生于 1980 年左右的 S 语言的一个分支。 R 语言是 S 语言的一种实现。S 语言是由 AT&T 贝尔实验室开发的一种用来进行数据探索、统计分析、作图的解释型语言。

R 是一套完整的数据处理、计算和制图软件系统。其功能包括：数据存储和处理系统；数组运算工具（其向量、矩阵运算方面功能尤其强大）；完整连贯的统计分析工具；优秀的统计制图功能；简便而强大的编程语言：可操纵数据的输入和输入，可实现分支、循环，用户可自定义功能。

与其说 R 是一种统计软件，还不如说 R 是一种数学计算的环境，因为 R 并不是仅仅提供若干统计程序、使用者只需指定数据库和若干参数便可进行一个统计分析。R 的思想是：它可以提供一些集成的统计工具，但更大量的是它提供各种数学计算、统计计算的函数，从而使使用者能灵活机动的进行数据分析，甚至创造出符合需要的新的统计计算方法。

R 是一个免费的自由软件，它有 UNIX、LINUX、MacOS 和 WINDOWS 版本，都是可以免费下载和使用的。在 R 主页那儿可以下载到 R 的安装程序、各种外挂程序和文档。在 R 的安装程序中只包含了 8 个基础模块，其他外在模块可以通过 CRAN 获得。

回归分析

是揭示变量之间数量上的联系方式和变化规律的一种数理统计方法。在社会学研究中常用这种方法测定某一社会因素对另一社会因素的影响程度，并从一个(或多个)社会因素的变化推测另一个社会因素的发展趋势。回归分析的内容包括通过确定变量间的关系建立回归方程，对回归方程的可信程度进行统计检验，利用回归方程对变量进行预测，以及从影响一个变量的许多变量中判断各变量影响的显著程度。回归方程按涉及变量的多少可分为一元回归方程和多元回归方程。一元回归方程表达式为：

$$\hat{y}=a+bx$$

$$b=\frac{n\sum xiyi-\sum xiyi}{n\sum x^2i-(\sum xi)^2} \quad a=\bar{y}-b\bar{x}$$

式中 a 为方程的常数项;b 为回归系数; y 为预测值;x_i 与 y_i 分别为自变量和依变量。这一方程为 y 变量对 x 变量的回归方程。
-- 社会科学大词典

回归分析　　Регрессионный анали

统计分析的一种， 它能弄清楚结果特征 (解释变数)变化的平均值与一个或数个特征因素(解释变数) 的依存关系。回归线是指按第一特征的平均值(这个值对应于特征因素的平均区间) 而划的一条线。

当研究课题要求弄清楚一个特征的平均值在第二特征(或几个特征关系)变化为一时有多大变化，这时可使用回归分析。在社会学研究中回归分析可用来研究劳动生产率的社会因素，研究影响工作时间、自由时间、以及工作成绩等因素。回归分析能确定提高一级机床工人的文化水平， 就要增长平均产量 4%。回归分析只可用于数量变数的研究(被测的区间及变数) ， 回归分析本身包括回归方程、评估及其分析方程在内。

1.建立回归方程。方程的基本阶段(步骤)如下(如我们探讨一对回归， 即探讨描述一个特征因素影响结果特征的方程)： 1) 联系方式的初步计划。在不大的子样本中， 用图似法划出一条回归线。这种线性关系写成方程 yx=a+bx， 其中 a 表示方程的项， 对方程中未考虑的因素的结果特征起作用， 方程中的 b 表示回归系数; 2)计算回归方程的参数(a 和 b 叫做方程的参数) 。为了求出参数必需解两个方程组：

$$\begin{cases} an+b\Sigma_x=\Sigma_y \\ a\Sigma_x+b\Sigma_x^2=\Sigma_{yx} \end{cases}$$

解方程组得出计算参数的下列公式：

$$a=\frac{\Sigma_y\Sigma_x^2-\Sigma_{yx}\Sigma_x}{n\Sigma_{x2}-\Sigma_x\Sigma_x}; \quad b=\frac{n\Sigma_{xy}-\Sigma_x\Sigma_y}{n\Sigma_{x2}-\Sigma_x\Sigma_x}$$

例：设 x 为业务员的工作速度;y 表示工作质量即误差数。假设

我们对下面五个业务员的工作进行测定，即：

编号	工作速度	工作质量	c^2	$x \cdot y$	计算
1	$x_1 = 5$	$y_1 = 3$	25	15	$\begin{cases} 5a + 26b = 21 \\ 26a + 114b = 112 \end{cases}$
2	$x_2 = 7$	$y_2 = 5$	49	35	$a = \dfrac{21 \times 144 - 112 \times 26}{5 \times 144 - 26 \times 26} = 2.54$
3	$x_3 = 3$	$y_3 = 4$	9	12	
4	$x_4 = 6$	$y_4 = 5$	36	30	
5	$x_5 = 5$	$y_5 = 4$	25	20	$b = \dfrac{5 \times 12 - 26 \times 21}{5 \times 144 - 26 \times 26} = 0.32$
$n = 5$	$\Sigma_x = 26$	$\Sigma_y = 21$	$\Sigma^2_{xi} = 144$	$\Sigma_{xy} = 112$	

将 a、b 值代入方程 y=a+bx，即得回归方程 y=2.54+0.32x。

2.方程评价。为检验计算的准确性我们将

$$\bar{x} = \frac{\Sigma xi}{n} = \frac{26}{5} = 5.2$$

代入所得方程。如果求得的参数准确，则得：

$$y = \frac{\Sigma yi}{n} = \frac{21}{5} = 4.2$$

$$\bar{y} = a + \bar{b}\bar{x} = 2.54 + 0.32 \times 5.2 = 4.2$$

把 x_1 值代入方程，我们会求出计算值(理论值)y_i(用 $\hat{y}_i$ 表示)。在我们所举的例子中 $\hat{y}_1 = 4.14$；$\hat{y}_2 = 4.78$；$\hat{y}_3 = 3.5$；$\hat{y}_4 = 4.46$；$\hat{y}_5 = 4.14$。

可见，这些值近似经验值 yi。为了更精确的作出评价应求剩余方差 S^2 剩余 $=\frac{\Sigma(yi-\overline{yi})^2}{n}$，方差可表示成评价所求方程的未知判据；而方程与经验数据 $\hat{yi}$ 近似。如果计算值 $\hat{yi}$ 等于经验值 yx，那么剩余方差等于 0。

3.回归方程分析。回归系数 b=0.32 表明在增加单位时间、工作速度时， 误差平均应增加 0.32。

方程的自由项 a=2.54 表明， 平均来看，2.5 个误差是方程中评估得到的结果。既然误差的平均数 y=4.2，那么这种情况就占 60%多， 剩下 40%左右是估计得到的因素。从分析中得出的实际结论是： 工作速度的调节几乎不会涉及到三分之二的可能性错误。应该分析一下对如工作条件、熟练水平、刺激作用等这些因素对质量所产生的影响。

工厂社会学家在研究质量经济等类似课题时，常常产生一些在上例中所探讨的类似问题。小样本 (班组、职业团体、工段)运用回归分析能有效地完成工作， 并根据报告材料(不是问卷调查) 来制定有根据的建议。

在进行大量观察时(n>30)， 手工计算回归方程的参数劳动量很大。在评价结果特征与某些因素特征的关系时要作出多重回归方程。

为避免错误和节约时间， 计算多重回归方程参数必须用电子计算机进行。这种情况下， 系数的解释和成对回归情况下一样。

字数：1217

回归分析概念及原理

回归分析定义

利用数据统计原理，对大量统计数据进行数学处理，并确定因变量与某些自变量的相关关系，建立一个相关性较好的回归方程（函数表达式），并加以外推，用于预测今后的因变量的变化的分析方法

回归分析分类

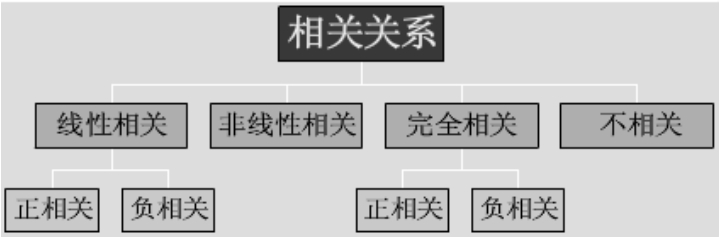
根据因变量和自变量的个数来分类：一元回归分析，多元回归分析

根据因变量和自变量的函数表达式来分类：线性回归分析，非线性回归分析

几点说明

- 通常情况下，线性回归分析是回归分析法中最基本的方法，当遇到非线性回归分析时，可以借助数学手段将其化为线性回归；因此，主要研究线性回归问题，一点线性回归问题得到解决，非线性回归也就迎刃而解了
- 在社会经济现象中，很难确定因变量和自变量之间的关系，它们大多是随机性的，只有通过大量统计观察才能找出其中的规律。随机分析是利用统计学原理来描述随机变量相关关系的一种方法

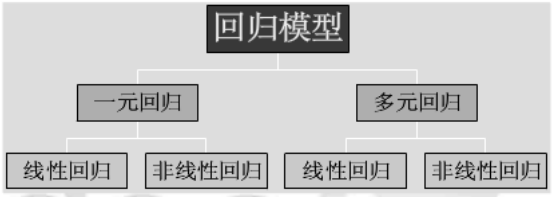
- 由回归分析法的定义知道，回归分析可以简单的理解为信息分析与预测。信息即统计数据，分析即对信息进行数学处理，预测就是加以外推，也就是适当扩大已有自变量取值范围，并承认该回归方程在该扩大的定义域内成立，然后就可以在该定义域上取值进行"未来预测"。当然，还可以对回归方程进行有效控制
- 相关关系：可以分为确定关系和不确定关系。但是不论是确定关系或者不确定关系，只要有相关关系，都可以选择一适当的数学关系式，用以说明一个或几个变量变动时，另一个变量或几个变量平均变动的情况



回归分析主要解决的问题

回归分析主要解决方面的问题

- 确定变量之间是否存在相关关系，若存在，则找出数学表达式
- 根据一个或几个变量的值，预测或控制另一个或几个变量的值，且估计这种控制或预测可以达到何种精确度



回归分析步骤

1. 根据自变量与因变量的现有数据以及关系，初步设定回归方程
2. 求出合理的回归系数
3. 进行相关性检验，确定相关系数
4. 在符合相关性要求后，即可根据已得的回归方程与具体条件相结合，来确定事物的未来状况，并计算预测值的置信区间

回归分析的有效性和注意事项

有效性

- 用回归分析法进行预测首先要对各个自变量做出预测。若各个自变量可以由人工控制或易于预测，而且回归方程也较为符合实际，则应用回归预测是有效的，否则就很难应用

注意事项

- 为使回归方程较能符合实际，首先应尽可能定性判断自变量的可能种类和个数，并在观察事物发展规律的基础上定性判断回归方程的可能类型；其次，力求掌握较充分的高质量统计数据，再运用统计方法，利用数学工具和相关软件从定量方面计算或改进性判断

常用概念

实际值

- 实际观测到的研究对象特征数据值

理论值

- 根据实际值我们可以得到一条倾向线，用数学方法拟合这条曲线，可以得到数学模型，根据这个数学模型计算出来的、与实际值相对应的值，称为理论值

表示符号

- 实际值，用y_i表示；理论值，用$\hat{y}_i$表示；预测值，用y_0表示。

一元线性回归

一元线性回归，就是只涉及一个自变量的回归；自变量和因变量之间的关系是线性关系的回归；因变量与自变量之间的关系用一条线性方程来表示的回归

方法步骤

1. 确定回归模型：

由于我们研究的是一元线性回归，因此其回归模型可表示为：$y=\beta_0+\beta_1 x+\varepsilon$；

其中，y是因变量；x是自变量；ε是误差项；β_0和β_1称为模型参数（回归系数）。

2. 求出回归系数：

这里的回归系数的求解，就要用一定的方法，使得该系数应用于该方程是“合理的”。最常用的一种方法就是最小二乘估计法。最小二乘法是测量工作和科学实验中最常用的一种数据处理方法，其基本原理是，根据实验观测得到的自变量 x 和因变量 y 之间的一组对应关系，找出一个给定类型的函数$y=f(x)$，使得它所取的值$f(x_1),f(x_2),\cdots,f(x_n)$与观测值$y_1,y_2,\cdots,y_n$在某种尺度下最接近，即在各点处的偏差的平方和达到最小，即$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2=\sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1 x_i)^2=$最小。这种方法求的的$\hat{\beta}_0$和$\hat{\beta}_1$将使得拟合直线$y=\hat{\beta}_0+\hat{\beta}_1 x$中的$y$和$x$之间的关系与实际数据的误差比其他任何直线都小。

根据最小二乘法的要求，可以推导得到最小二乘法的计算公式：

$$\begin{cases}\hat{\beta}_1=\dfrac{n\sum_{i=1}^{n}x_iy_i-\left(\sum_{i=1}^{n}x_i\right)\left(\sum_{i=1}^{n}y_i\right)}{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2} \\ \hat{\beta}_0=\bar{y}-\hat{\beta}_1\bar{x}\end{cases}$$ 其中，$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i,\bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$；

相关性检验：

对于若干组具体数据(x_i,y_i)都可算出回归系数$\hat{\beta}_0,\hat{\beta}_1$，从而得到回归方程。至于$y$与$x$之间是否真有如回归模型所描述的关系，或者说用所得的回归模型去拟合实际数据是否有足够好的近似，并没有得到判明。因此，必须对回归模型描述实际数据的近似程度，也即对所得的回归模型的可信程度进行检验，称为相关性检验。

相关系数是衡量一组测量数据x_i,y_i线性相关程度的参量，其定义为：

$$r=\frac{\overline{xy}-\bar{x}\bar{y}}{\sqrt{(\overline{x^2}-\bar{x}^2)(\overline{y^2}-\bar{y}^2)}}$$，或者$$r=\frac{n\sum x_iy_i-\sum x_i\sum y_i}{\sqrt{[n\sum_{i=1}^{n}x_i^2-\sum_{i=1}^{n}x_i^2][n\sum_{i=1}^{n}y_i^2-\sum_{i=1}^{n}y_i^2]}}$$

r值在$0<|r|\leqslant 1$中。$|r|$越接近于1，x,y之间线性好；r为正，直线斜率为正，称为正相关；r为负，直线斜率为负，称为负相关。$|r|$接近于0，则测量数据点分散或x_i,y_i之间为非线性。不论测量数据好坏都能求出$\hat{\beta}_0$和$\hat{\beta}_1$，所以我们必须有一种判断测量数据好坏的方法，用来判断什么样的测量数据不宜拟合，判断的方法是$|r|<r_0$时，测量数据是非线性的．r_0称为相关系数的起码值，与测量次数n有关，如下表：

相关系数起码值r_0

n	r_0	n	r_0	n	r_0
3	1.000	9	0.798	15	0.641
4	0.990	10	0.765	16	0.623
5	0.959	11	0.735	17	0.606
6	0.917	12	0.708	18	0.590
7	0.874	13	0.684	19	0.575
8	0.834	14	0.661	20	0.561

在进行一元线性回归之前应先求出r值，再与r_0比较，若$|r|>r_0$，则x和y具有线性关系，可求回归直线；否则反之。

置信区间的确定：

当确定相关性后，就可以对置信区间进行确定，就可以结合实际情况，确定事物未来的状况了。回归分析的最主要的应用就在于“预测”，而预测是不是准确的，就得有一个衡量的工具。它就是置信区间。或者从另外一方面来说，回归方程是

由数理统计得出的，它反映的是实际数据的统计规律，所以，根据回归方程所得的预测值 y_0 只是对应于 x_0 的单点预测估计值，预测值应该有一个置信区间。这样来看，计算置信区间就是很有必要的。
置信区间：

$S^2=\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n-2}$，其中 S^2 是 σ^2 的无偏估计量，S^2 称为剩余方差，S 称为剩余标准差。[注：该表达式的自由度为 $n-2$ 是因为有 2 个限制变量 x_i 和 y_i]故对于给定的 x_0，y 值的概率为 0.95 的置信区间是：$(y_0-1.96S, y_0+1.96S)$。点击参看置信区间的确定内容。

回归问题分析的方法

- 收集数据
- 建立一个适合相关问题的回归函数
- 通过学习已有的数据，对函数的未知参数进行估计
- 回归方程及回归系数的显著性检验
- 利用这个模型去预测/分类新的数据

回归分析的原理

目的是为了找出一个最能够代表所有观测资料的函数，用这个函数来表示因变量与自变量的关系。回归的基本模型一般可以表示为：Y=f(X ,u)，其中 Y 为因变量，X 为自变量，u 为未知参数

回归分析方法分类

根据用途可以有多种分类，单变量的线性回归模型，多变量的线性回归模型，广义线性回归模型，概率单位回归模型，逻辑回归模型，曲线回归模型，岭回归模型，主成分回归模型等

进步是留给时间最美的礼物

Python 回归分析五部曲（一）—简单线性回归

回归最初是遗传学中的一个名词，是由英国生物学家兼统计学家高尔顿首先提出来的，他在研究人类身高的时候发现：高个子回归人类的平均身高，而矮个子则从另一方向回归人类的平均身高；

回归分析整体逻辑

- 回归分析（Regression Analysis）
  - 研究自变量与因变量之间关系形式的分析方法，它主要是通过建立因变量 y 与影响它的自变量 x_i(i=1,2,3… …)之间的回归模型，来预测因变量 y 的发展趋向。
- 回归分析的分类
  - 线性回归分析
  - 简单线性回归
  - 多重线性回归

    - 非线性回归分析
    - 逻辑回归
    - 神经网络
- 回归分析的步骤
    - 根据预测目标，确定自变量和因变量
    - 绘制散点图，确定回归模型类型
    - 估计模型参数，建立回归模型
    - 对回归模型进行检验
    - 利用回归模型进行预测

简单线性回归模型

1.基础逻辑

$$y=a+bx+e$$

该模型也称作一元一次回归方程，模型中：

- y：因变量
- x：自变量
- a：常数项（回归直线在 y 轴上的截距）
- b：回归系数（回归直线的斜率）
- e：随机误差（随机因素对因变量所产生的影响）
    - e 的平方和也称为残差，残差是判断线性回归拟合好坏的重要指标之一

从简单线性回归模型可以知道，简单线性回归是研究一个因变量与一个自变量间线性关系的方法

2.案例实操

下面我们来看一个案例，某金融公司在多次进行活动推广后记录了活动推广费用及金融产品销售额数据，如下表所示

序号	活动推广费	销售额
1	19	60
2	45	113
3	35	94
4	31	90
5	25	60
6	32	88
7	21	59
8	26	61
9	24	57
10	27	78
11	9	27
12	23	72
13	33	85
14	29	63

因为活动推广有明显效果，现在的需求是投入 60 万的推广费，能得到多少的销售额呢？这时我们就可以使用简单线性回归模型去解决这个问题，下面，我们用这个案例来学习，如何进行简单线性回归分析；

（1）第一步 确定变量

- 根据预测目标，确定自变量和因变量
  - 问题：投入 60 万的推广费，能够带来多少的销售额？
  - 确定因变量和自变量很简单，谁是已知，谁就是自变量，谁是未知，就就是因变量，因此，推广费是自变量，销售额是因变量；

```
import numpy
from pandas import read_csv
from matplotlib import pyplot as plt
from sklearn.linear_model import LinearRegression

data = read_csv(
    'file:///Users/apple/Desktop/jacky_1.csv',encoding='GBK'
```

)

（2）第二步 确定类型

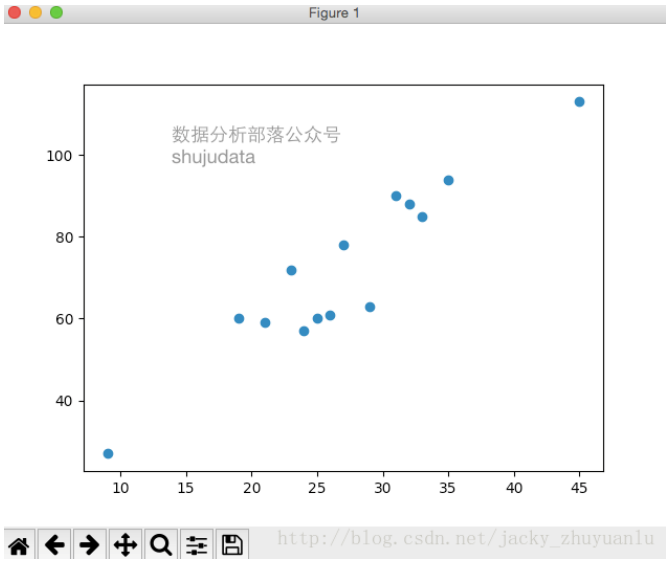
- 绘制散点图，确定回归模型类型
  - 根据前面的数据，画出自变量与因变量的散点图，看看是否可以建立回归方程，在简单线性回归分析中，我们只需要确定自变量与因变量的相关度为强相关性，即可确定可以建立简单线性回归方程，根据 jacky 前面的文章分享《Python 相关分析》，我们很容易就求解出推广费与销售额之间的相关系数是 0.94，也就是具有强相关性，从散点图中也可以看出，二者是有明显的线性相关的，也就是推广费越大，销售额也就越大

```
#画出散点图，求 x 和 y 的相关系数
plt.scatter(data.活动推广费,data.销售额)

data.corr()
```

```
/Library/Frameworks/Python.Framework/Versions/3.5/bin/python3.5 /use
  .ipynb_checkpoints/111.py
            序号      活动推广费      销售额
序号        1.000000 -0.297891 -0.393672
活动推广费 -0.297891  1.000000  0.941814
销售额     -0.393672  0.941814  1.000000
```

数据分析部落
公众号
shujudata



（3）第三步 建立模型

- 估计模型参数，建立回归模型

  要建立回归模型，就要先估计出回归模型的参数 A 和 B，那么如何得到最佳的 A 和 B，使得尽可能多的数据点落在或者更加靠近这条拟合出来的直线上呢？

统计学家研究出一个方法，就是最小二乘法，最小二乘法又称最小平方法，通过最小化误差的平方和寻找数据的最佳直线，这个误差就是实际观测点和估计点间的距离；
最小二乘法名字的缘由有二个：一是要将误差最小化，二是使误差最小化的方法是使误差的平方和最小化；在古汉语中，平方称为二乘，用平方的原因就是要规避负数对计算的影响，所以最小二乘法在回归模型上的应用就是要使得实际观测点和估计点的平方和达到最小，也就是上面所说的使得尽可能多的数据点落在或者说更加靠近这条拟合出来的直线上；
我们只要了解最小二乘法的原理即可，具体计算的过程就交给 Python 处理。


```
#估计模型参数，建立回归模型
'''
(1) 首先导入简单线性回归的求解类 LinearRegression
(2) 然后使用该类进行建模，得到 lrModel 的模型变量
'''
lrModel = LinearRegression()
#(3) 接着，我们把自变量和因变量选择出来
x = data[['活动推广费']]
y = data[['销售额']]
#模型训练
'''
调用模型的 fit 方法，对模型进行训练
这个训练过程就是参数求解的过程
并对模型进行拟合
'''
lrModel.fit(x,y)
```

（4）第四步 模型检验

- 对回归模型进行检验
  - 回归方程的精度就是用来表示实际观测点和回归方程的拟合程度的指标，使用判定系数来度量。

判定系数＝相关系数 $R^2=ESSTSS=1-RSSTSS$

其中，数据分析部落公众号(shujudata)

$TSS=\sum(Y_i-\bar{Y})^2$ 总离差平方和

$$ESS=\sum(Y_i-\bar{Y})^2$$ 回归差平方和

$$RSS=\sum(Y_i-\bar{Y})^2$$ 残差平方和

解释：判定系数等于相关系数 R 的平方用于表示拟合得到的模型能解释因变量变化的百分比，R 平方越接近于 1，表示回归模型拟合效果越好

如果拟合出来的回归模型精度符合我们的要求，那么我们可以使用拟合出来的回归模型，根据已有的自变量数据来预测需要的因变量对应的结果

```
#对回归模型进行检验
lrModel.score(x,y)
```

执行代码可以看到，模型的评分为 0.887,是非常不错的一个评分，我们就可以使用这个模型进行未知数据的预测了

（5）第五步 模型预测

- 调用模型的 predict 方法，这个就是使用 sklearn 进行简单线性回归的求解过程；

```
lrModel.predict([[60],[70]])
```

- 如果需要获取到拟合出来的参数各是多少，可以使用模型的 intercept 属性查看参数 a(截距)，使用 coef 属性查看参数 b

```
#查看截距
alpha = lrModel.intercept_[0]
#查看参数
beta = lrModel.coef_[0][0]

alpha + beta*numpy.array([60,70])
```

3.完整代码

```
import numpy
from pandas import read_csv
from matplotlib import pyplot as plt
from sklearn.linear_model import LinearRegression

data = read_csv(
    'file:///Users/apple/Desktop/jacky_1.csv',encoding='GBK'
)
#画出散点图，求 x 和 y 的相关系数
plt.scatter(data.活动推广费,data.销售额)
data.corr()
#估计模型参数，建立回归模型
'''
```

```
(1) 首先导入简单线性回归的求解类 LinearRegression
(2) 然后使用该类进行建模，得到 lrModel 的模型变量
'''
lrModel = LinearRegression()
#(3) 接着，我们把自变量和因变量选择出来
x = data[['活动推广费']]
y = data[['销售额']]
#模型训练
'''
调用模型的 fit 方法，对模型进行训练
这个训练过程就是参数求解的过程
并对模型进行拟合
'''
lrModel.fit(x,y)
#对回归模型进行检验
lrModel.score(x,y)
#利用回归模型进行预测
lrModel.predict([[60],[70]])
#查看截距
alpha = lrModel.intercept_[0]
#查看参数
beta = lrModel.coef_[0][0]
alpha + beta*numpy.array([60,70])
```

4.总结-sklearn 建模流程

- sklearn 建模流程
  - 建立模型
  - lrModel = sklearn.linear_model.LinearRegression()
  - 训练模型
  - lrModel.fit(x,y)
  - 模型评估
  - lrModel.score(x,y)

- 模型预测
- lrModel.predict(x)

Python 回归分析五部曲（二）—多重线性回归

基础铺垫

- 多重线性回归（Multiple Linear Regression）
  - 研究一个因变量与多个自变量间线性关系的方法
  - 在实际工作中，因变量的变化往往受几个重要因素的影响，此时就需要用 2 个或 2 个以上的影响因素作为自变量来解释因变量的变化，这就是多重线性回归;

多重线性回归模型

1.模型

$$y=\alpha+\beta_1x_1+\beta_2x_2+...+\beta_nx_n+e$$

数据分析部落公众号：shujudata

方程式中：

y–因变量

x_n–第 n 个自变量

α–常数项（回归直线在 y 轴上的截距）

β_n–第 n 个偏回归系数

e–随机误差

2.模型关键词解析

- 偏回归系数
  - 多重线性模型中包含多个自变量，它们同时对因变量 y 发生作用，如果要考察一个自变量对因变量 y 的影响，就必须假设其他自变量保持不变；因此，多重线性模型中的回归系数称为偏回归系数，偏回归系数 β_1 是指在其他自变量保持不变的情况下，自变量 x_1 每变动一个单位，引起的因变量 y 的平均变化；β_2 到 β_n 依次类推；

回顾－回归分析步骤

- 根据预测目标，确定自变量和因变量
- 绘制散点图，确定回归模型类型
- 估计模型参数，建立回归模型
- 对回归模型进行检验
- 利用回归模型进行预测

案例实操-金融场景

下面，jacky 通过一个金融场景的案例，开始我们的分享：某金融公司打算新开一类金融产品，现有 9 个金融产品的数据，包括用户购买金融产品的综合年化利率，以及公司收取用户的佣金（手续费）；如下表所示，产品利率为 11%，佣金为 50，我们需要预测这款金融产品的销售额

产品编号	百分比利率	抽取用户佣金	金融产品销售额
1	9	75	500
2	7	30	370
3	7	20	375
4	5	30	270
5	6	0	360
6	7	21	379
7	8	50	440
8	6	20	300
9	9	60	510
10	11	50	?

```
import pandas
data = pandas.read_csv(
    'file:///Users/apple/Desktop/jacky_1.csv',encoding='GBK'
)
```

第一步 确定变量

- 根据预测目标，确定自变量和因变量
  - 因变量：销售额
  - 自变量：利率、佣金

第二步 确定类型

- 绘制散点图，确定回归模型类型
  - 从散点图和相关系数结果表可以看出，产品利率和销售额是强正相关；佣金与销售额是强负相关；因此，我们可以使用多重线性模型来解决这个问题；

我们对自变量和因变量绘制散点图，因为需要绘制多个变量两两之间的散点图，在这里介绍一个更先进的绘图方法 scatter_matrix：我们把自变量和因变量从 data 中选取出来，然后设置好对应的参数。第一个是图片的大小，如果变量太多，我们就要把图片的尺寸设置的足够大才能够展示出来；第二个参数 diagonal 是变量与变量本身的绘图方式，我们选择 kde,是绘制直方图，这个参数是什么意思，我们执行代码就知道了，代码如下：

```
import matplotlib
from pandas.tools.plotting import scatter_matrix
font = {
    'family':'SimHei'
}
matplotlib.rc('font',**font)

scatter_matrix(
    data[["百分比利率","抽取用户佣金","金融产品销售额"],
    figsize =(10,10),diagonal = 'kid'
)
data[["百分比利率","抽取用户佣金","金融产品销售额"]].corr()
x = data[["百分比利率","抽取用户佣金"]]
y = data[["金融产品销售额"]]
```

第三步 建立模型

- 估计模型参数，建立回归模型
  - 多重线性回归模型参数的估计方法与简单线性回归模型参数的估计方法是相同的：都是采用最小二乘法进行估计（对最小二乘法更详细的解析，请参见 Python 回归分析五步曲（一）—简单线性回归）

```
#建模
from sklearn.linear_model import LinearRegression
lrModel = LinearRegression()
#训练模型
lrModel.fit(x,y)
```

第四步 模型检验

- 对回归模型进行检验

判定系数＝相关系数 $R^2=ESSTSS=1-RSSTSS$

调整判定系数＝相关系数 $R^{-2}=1-RSS/(n-k-1)TSS/(n-1)$

其中，数据分析部落公众号(shujudata)

$TSS=\sum(Y_i-Y^-)2$ 总离差平方和

$ESS=\sum(Y_i-Y^-)2$ 回归差平方和

$RSS=\sum(Y_i-Y^-)2$ 残差平方和

n 样本个数

k 自变量个数

- jacky 解析：拟合完方程的参数之后，我们就要对回归模型进行检验，在简单线性回归的分享中，我们用判定系数来验证方程的拟合程度，而在多重线性回归中，如果在模型中增加一个自变量，模型中 R 平方往往也会相应增加，这就会给我们一个错觉：要使得模型拟合的好，只要增加自变量即可。因此，使用判定系数 R 平方来验证方程的拟合程度是不够科学的，需要自变量个数进行修正和调整，也就是调整判定系数；以上，我们只要理解原理即可，公式记不住也不要紧，知道多重线性模型需要用调整判定系数来判定方程的拟合程度，会用 Python 看结果就可以了。

第五步 模型预测

- 利用回归模型进行预测
  - 根据已有的自变量数据，预测需要的因变量对应的结果

```
#预测
lrModel.predict([11,50])
#查看参数
lrModel.coef_
#查看截距
lrModel.intercept_
```

总结－完整代码

```
#---author:朱元禄---
import pandas
data = pandas.read_csv(
    'file:///Users/apple/Desktop/jacky_1.csv',encoding='GBK'
)
import matplotlib
from pandas.tools.plotting import scatter_matrix
font = {
    'family':'SimHei'
}
matplotlib.rc('font',**font)
```

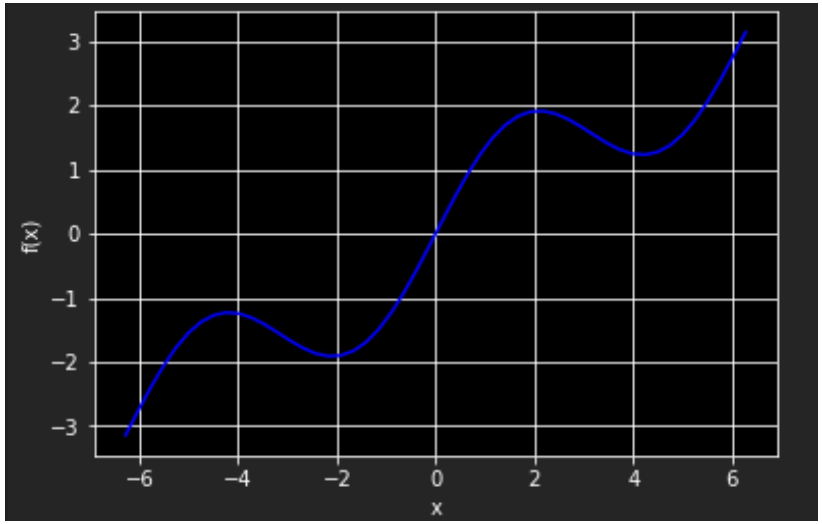
```
scatter_matrix(
    data[["百分比利率","抽取用户佣金","金融产品销售额"],
    figsize =(10,10),diagonal = 'kid'
)
data[["百分比利率","抽取用户佣金","金融产品销售额"]].corr()
x = data[["百分比利率","抽取用户佣金"]]
y = data[["金融产品销售额"]]
#建模
from sklearn.linear_model import LinearRegression
lrModel = LinearRegression()
#训练模型
lrModel.fit(x,y)
#预测
lrModel.predict([11,50])
#查看参数
lrModel.coef_
#查看截距
lrModel.intercept_
```

分类: python , 数据挖掘 , python

python 回归分析

假设原函数由一个三角函数和一个线性项组成

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
def f(x):
    return np.sin(x) + 0.5 * x
x = np.linspace(-2 * np.pi, 2 * np.pi, 50)
plt.plot(x, f(x), 'b')
plt.grid(True)
plt.xlabel('x')
plt.ylabel('f(x)')
```



一、用回归方式逼近

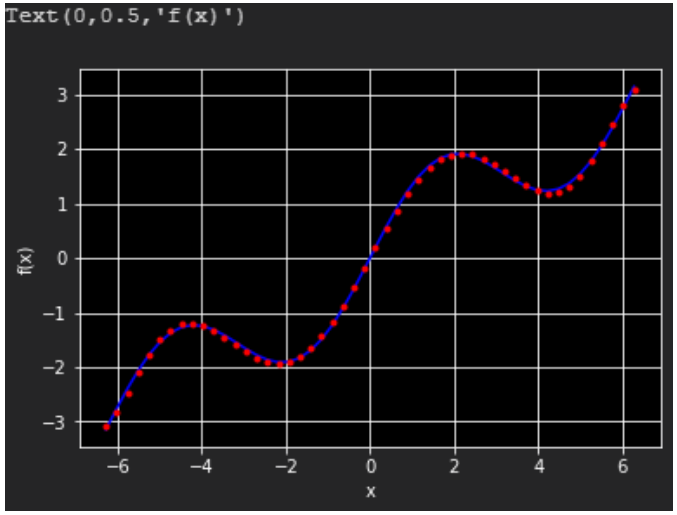
1. 作为基函数的单项式

最简单的情况是以单项式为基函数——也就是说，$b_1=1,b_2=x,b_3=x^2,b_4=x^3,...$ 在这种情况下，Numpy 有确定最优参数（polyfit）和以一组输入值求取近似值（ployval）的内建函数。ploy fit 函数参数如下：

参数	描述
x	x 坐标（自变量值）
y	y 坐标（因变量值）
deg	多项式拟合度
full	如果有真，返回额外的诊断信息
w	应用到 y 坐标的权重
cov	如果为真，返回协方差矩阵

典型向量化风格的 polyfit 和 polyval 线性回归（deg=7）应用方式如下：

```
reg = np.polyfit(x, f(x), deg=7)
ry = np.polyval(reg, x)
plt.plot(x,f(x),'b',label='f(x)')
plt.plot(x,ry,'r.',label='regression')
plt.grid(True)
plt.xlabel('x')
plt.ylabel('f(x)')
```



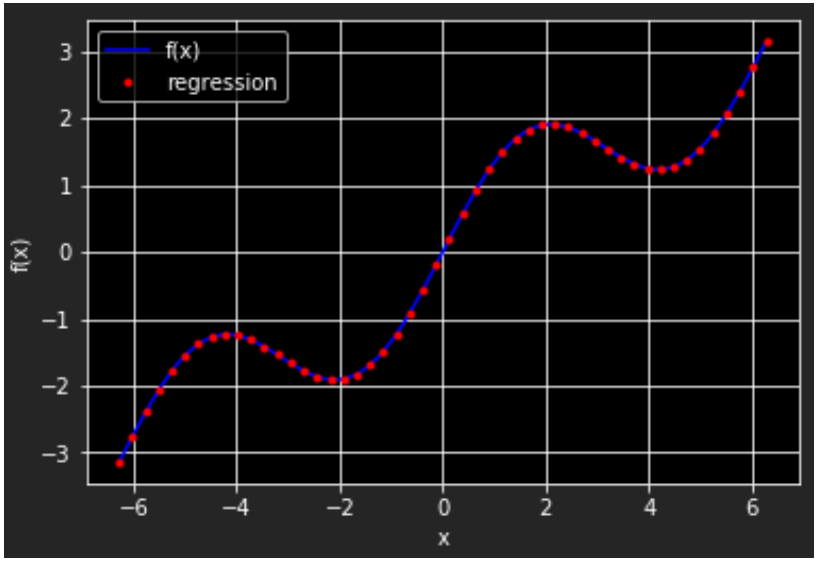
2. 单独的基函数

当选择更好的基函数组时，可以得到更好的回归结果。单独的基函数必须能通过一个矩阵方法定义（使用 Numpy ndarray 对象）。

numpy.linalg 子库提供 lstsq 函数，以解决最小二乘化做问题

```
matrix = np.zeros((3+1,len(x)))
matrix[3,:] = np.sin(x)
matrix[2,:] = x **2
matrix[1,:] = x
matrix[0,:] = 1
reg = np.linalg.lstsq(matrix.T,f(x))[0]
ry = np.dot(reg,matrix)
plt.plot(x,f(x),'b',label='f(x)')
plt.plot(x,ry,'r.',label='regression')
```

```
plt.legend(loc = 0)
plt.grid(True)
plt.xlabel('x')
plt.ylabel('f(x)')
```



对于有噪声的数据，未排序的数据，回归法都可处理。

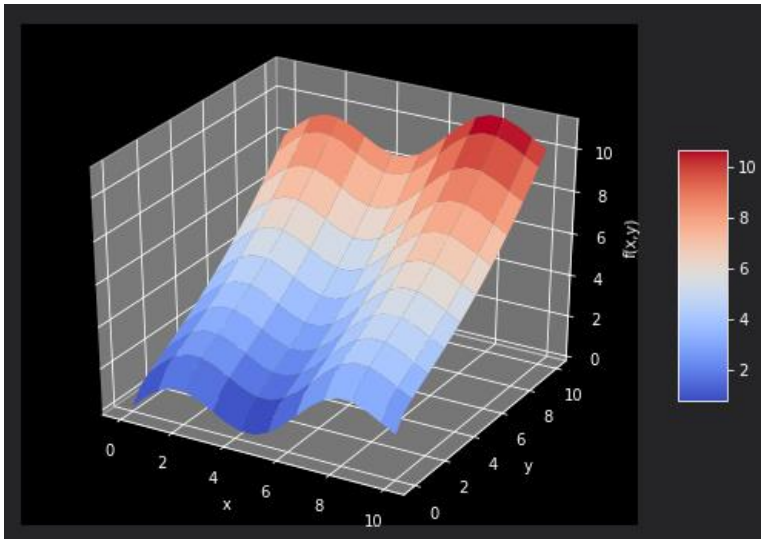
3. 多维

以 fm 函数为例

```
def fm(x,y):
    return np.sin(x) + 0.25 *x +np.sqrt(y) + 0.05 * y **2
x = np.linspace(0,10,20)
y = np.linspace(0,10,20)
X, Y = np.meshgrid(x,y)
Z = fm(X,Y)
x = X.flatten()
y = Y.flatten()

from mpl_toolkits.mplot3d import Axes3D
import matplotlib as mpl
```

```
fig = plt.figure(figsize=(9, 6))
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X,Y,Z,rstride=2,cstride=2,cmap=mpl.cm.coolwarm,linewidth=0.5,antialiased=True)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('f(x,y)')
fig.colorbar(surf,shrink=0.5,aspect=5)
```



为了获得好的回归结果，编译一组基函数，包括一个 sin 函数和 sqrt 函数。statsmodels 库提供相当通过和有益的函数 OLS，可以用于一维和多维最小二乘回归。

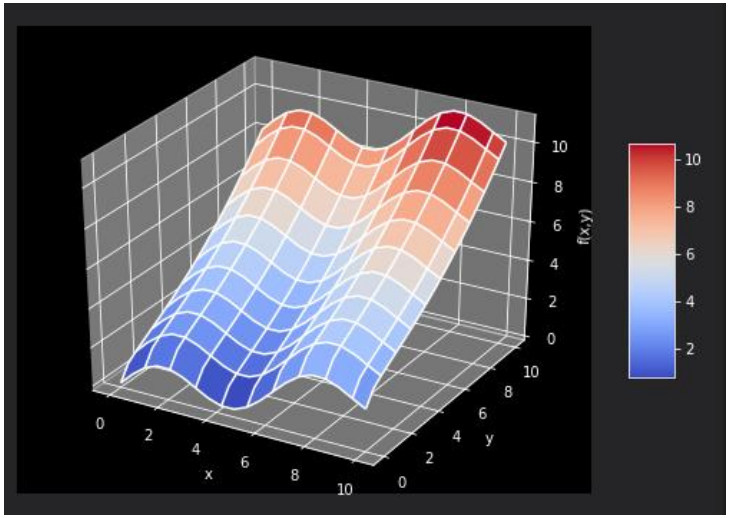
```
matrix = np.zeros((len(x),6+1))
matrix[:,6] = np.sqrt(y)
matrix[:,5] = np.sin(x)
matrix[:,4] = y ** 2
matrix[:,3] = x **2
matrix[:,2] = y
matrix[:,1] = x
matrix[:,0] = 1

import statsmodels.api as sm
model = sm.OLS(fm(x,y),matrix).fit()
```

OSL 函数的好处之一是提供关于回归及其而非结果的大量几百万来看信息。调用 model.summary 可以访问结果的一个摘要。单独统计数字（如确定系数）通常可以直接访问 model.rsquared。最优回归系数，保存在 model 对象的 params 属性中。

```
model.rsquared
a = model.params

def reg_func(a,x,y):
    f6 = a[6] * np.sqrt(y)
    f5 = a[5] * np.sin(x)
    f4 = a[4] * y ** 2
    f3 = a[3] * x ** 2
    f2 = a[2] * y
    f1 = a[1] * x
    f0 = a[0] * 1
    return (f6+f5+f4+f3+f2+f1+f0)
# reg_func 返回给定最优回归参数和自变量数据点的回归函数值
RZ = reg_func(a,X,Y)
fig = plt.figure(figsize=(9, 6))
ax = fig.gca(projection='3d')
surf1 = ax.plot_surface(X,Y,Z,rstride=2,cstride=2,cmap=mpl.cm.coolwarm,linewidth=0.5,antialiased=True)
surf2 = ax.plot_wireframe(X,Y,RZ,rstride=2,cstride=2,label = 'regression')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('f(x,y)')
fig.colorbar(surf,shrink=0.5,aspect=5)
```



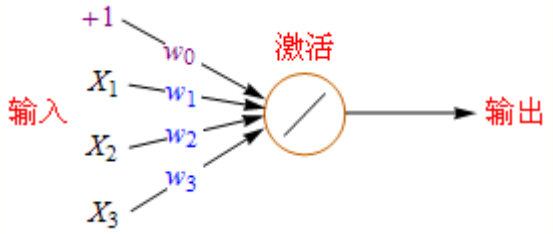
分类：Python

数据挖掘——回归分析 2——简单神经网络的 python 实现

神经网络(Artificial Neural Network)：全称为人工神经网络（ANN），是一种模仿生物神经网络（动物的中枢神经系统，特别是大脑）的结构和功能的数学模型或计算模型。

部分原理：

下面是单个神经元的数学模型：



+1 代表偏移值(偏置项, Bias Units)；X1,X2,X2 代表初始特征；w0,w1,w2,w3 代表权重(Weight)，即参数，是**特征的缩放倍数**；特征经过缩放和偏移后全部累加起来，此后还要经过一次**激活**运算然后再输出。

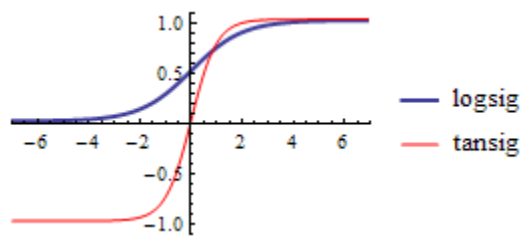
最常见的激活函数是 Sigmoid(S 形曲线)，Sigmoid 有时也称为逻辑回归(Logistic Regression)，简称 logsig。logsig 曲线的公式如下：

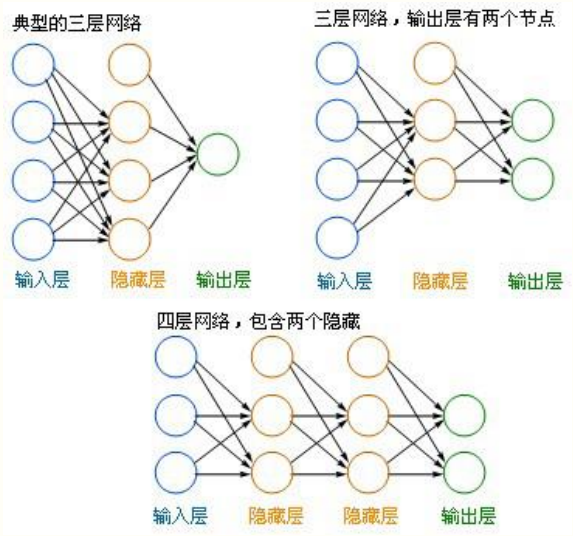
$$y = \text{logsig}(x) = \frac{1}{1+e^{-x}}$$

还有一种 S 形曲线也很常见到，叫双曲正切函数(tanh)，或称 tansig，可以替代 logsig。

$$y = \text{tansig}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

下面是它们的函数图形，从图中可以看出 logsig 的数值范围是 0~1，而 tansig 的数值范围是-1~1。





在 python 中的实现：

对训练集的预处理与逻辑回归相同，从模型构建开始不同

###对训练集做预处理操作

```
###模型构建、训练、评分
from sklearn.neural_network import MLPClassifier
for i in range(1,11):
    ANNmodel = MLPClassifier(
            activation='relu',    #激活函数为 relu,类似于 s 型函数
```

```
            hidden_layer_sizes=i)  #隐藏层为 i
    ANNmodel.fit(inputData,outputData)  #训练模型
    score = ANNmodel.score(inputData,outputData)  #模型评分
    print(str(i) + ',' + str(score))  #每次循环都打印模型评分
#模型评分基本稳定在 0.83x 左右
```

可以发现，隐藏层增大，模型评分趋于一个较稳定的值，即**并非隐藏层越多，模型越好。**

###对测试集做相同的预处理操作

```
###输入测试集作为参数
inputNewData = dummyNewData[dummySelect]
###得到预测结果，以序列形式进行输出
ANNmodel.predict(inputNewData)
```