回归问题

线性回归

线性回归

1. **目的**：用于预测连续的输出值。线性回归是用来估计因变量（目标变量）和一个或多个解释变量（特征）之间关系的线性模型。
2. **函数形式**：线性回归的基本形式是 $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$，其中 Y 是因变量，X_i 是解释变量，β 是系数，而 ϵ 是误差项。
3. **输出**：输出是一个实数，可以是任何值。
4. **适用场景**：用于预测或估计一个数量（如销售额、温度、收入等）。
5. **假设**：线性关系、同方差性、误差项的独立性和正态分布等。

总的来说，线性回归是根据已有的数据集，包括自变量和因变量，我们可以通过线性回归预测出一个函数来模拟他们之间的关系，之后有新的自变量输入时就可以预测出它的因变量

逻辑回归

逻辑回归

1. **目的**：用于预测分类的输出值。逻辑回归是用来估计因变量与一个或多个解释变量之间关系的分类模型，通常用于二分类问题。
2. **函数形式**：逻辑回归使用的是对数几率作为链接函数，基本形式是 $\log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$，其中 p 是属于特定类别的概率。
3. **输出**：输出是一个概率值，范围在0到1之间，表示特定类别的概率。
4. **适用场景**：用于预测一个事件发生与否（如疾病的发生、电子邮件是否为垃圾邮件等）。
5. **假设**：逻辑回归不要求因变量和解释变量之间有线性关系，也不需要误差项的同方差性和正态分布。

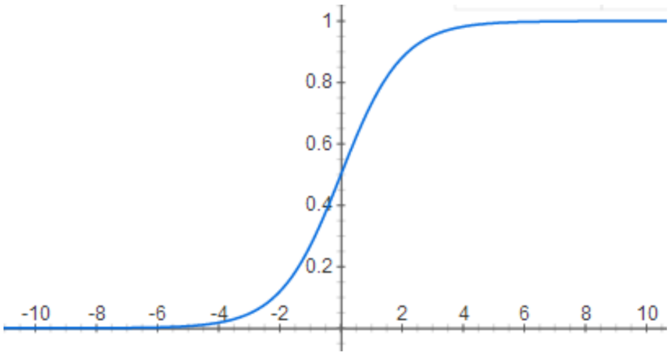
逻辑回归相当于是分类classification问题，首先将变量的输出归一到0到1之间，则大于0.5输出的就是1，小于0.5输出的就是0，通过sigmoid函数来实现

sigmoid函数：

逻辑斯蒂回归将线性回归拟合函数的输出通过Sigmoid函数映射到概率区间[0,1]，利用概率构造二分类模型（实际上是分类模型）

线性回归函数： $z = \theta^T x$

Sigmoid函数： $g(z) = \frac{1}{1+e^{-z}}$



逻辑斯蒂拟合函数(Hypothesis Function)： $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

之后我们要去设定cost function代价函数

该式可写为：

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1-y)log(1-h_\theta(x))$$

可以看出，这样的代价函数在y=1的前提下，如果预测值越接近1，那么相应代价就越小，反之则越大（对于y=0亦然）

设m为样本数：

将所有m个样本的代价累加并平均，我们有最终的代价函数：

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))]$$

---

cost function可用于梯度下降，因为sigmoid函数是非凸的，容易陷入局部最小值，所以梯度下降的步骤不可以缺少

实例

假设我们有一个数据集，包括学生的两个特征：学习时间（小时）和上课出勤率（%），以及他们是否通过了考试（通过=1，未通过=0）。我们的目标是建立一个逻辑回归模型来预测学生是否会通过考试。

示例数据集

学生	学习时间（小时）	出勤率（%）	是否通过考试（1=通过，0=未通过）
A	10	90	1
B	8	75	1
C	3	50	0
D	2	60	0
E	5	70	1

1. **定义模型**：
   $P(Y=1)=\frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2)}}$
   其中，X_1 是学习时间，X_2 是出勤率，β_0,β_1,β_2 是模型参数。
2. **初始化参数**：
   假设初始参数 β_0,β_1,β_2 都是0。
3. **计算预测概率**：
   对每个学生，将他们的学习时间和出勤率代入模型，计算预测概率。例如，对学生A：
   $P(Y=1)=\frac{1}{1+e^{-(0+0\times 10+0\times 90)}}=0.5$
   初始时，所有预测概率都是0.5。
4. **计算损失函数**：
   使用对数似然损失函数来计算损失。对于单个样本：
   $L(\beta)=-[y\log(P)+(1-y)\log(1-P)]$
   对于整个数据集，计算所有样本损失的平均值。
5. **梯度下降**：
   使用梯度下降算法来更新参数 β_0,β_1,β_2。对每个参数，计算损失函数关于该参数的偏导数，然后更新参数：
   $\beta_j:=\beta_j-\alpha\frac{\partial L}{\partial \beta_j}$
   其中，α 是学习率。
6. **重复迭代**：
   重复步骤3到5，直到模型收敛（即参数变化非常小或达到预设的迭代次数）。

通常来说有n个变量就有n+1个变换参数，一开始的定义模型就是将

$y=\beta_0+\beta_1X_1+\beta_2X_2$带入到$P(Y=1)=\frac{1}{1+e^y}$中，得到了初始的模型函数，注意，这里的参数初始都为0，后面会慢慢更新，则一开始算出来的预测概率都是0.5，之后我们可以用对数似然损失函数来计算损失，得到L也是$J(\beta_0,\beta_1,\beta_2)$，之后用梯度下降的方法来更新$\beta_0,\beta_1,\beta_2$。重复上述步骤得到最佳参数，这个时候参数变化十分小。