****

****

**QG中期考核详细报告书**

**题    目**

**学   院**

**专 业**

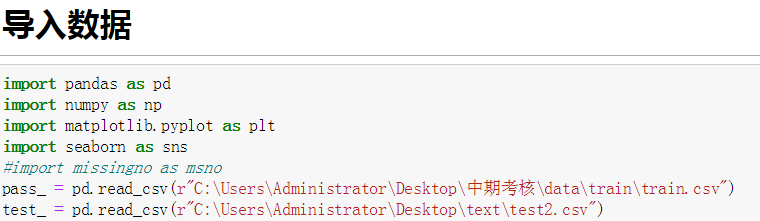
**年级班别**

**学 号**

**学生姓名**

**年 月 日**

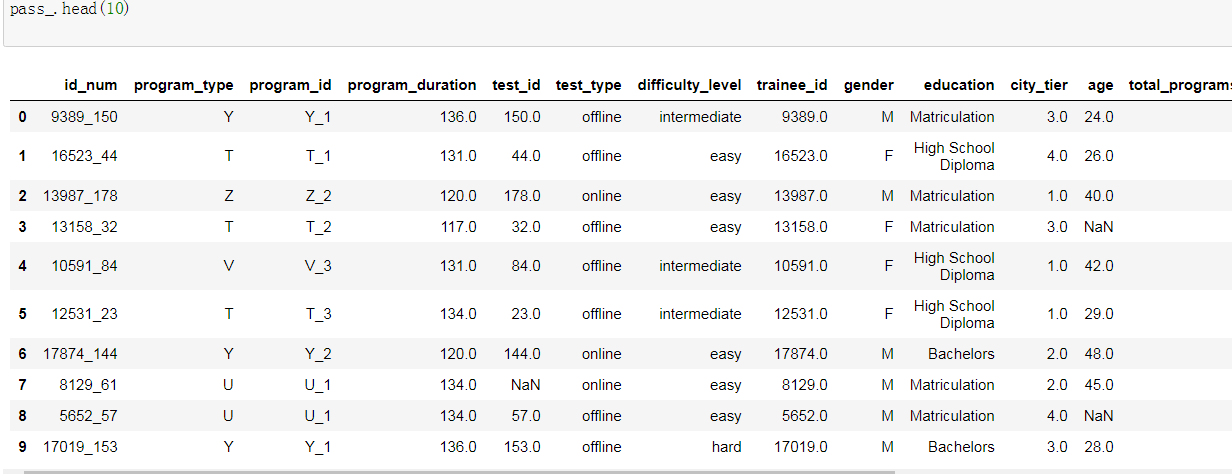
1 导入数据与相关库



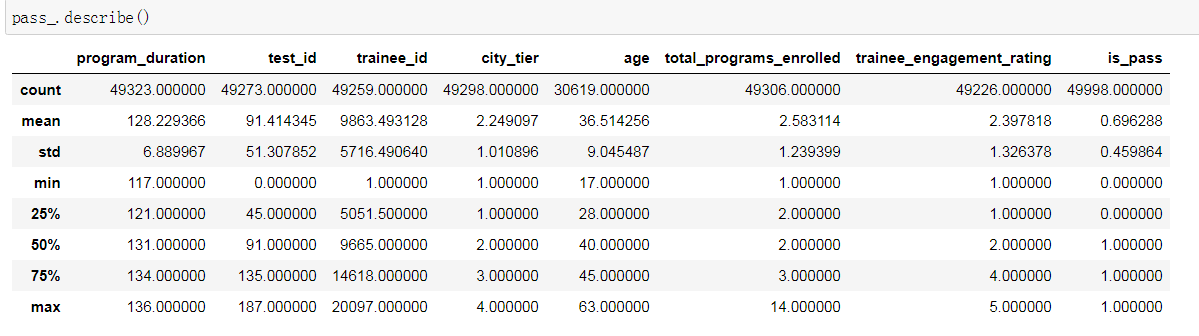
导入csv文件 pd.read\_csv

2 查看相关数据

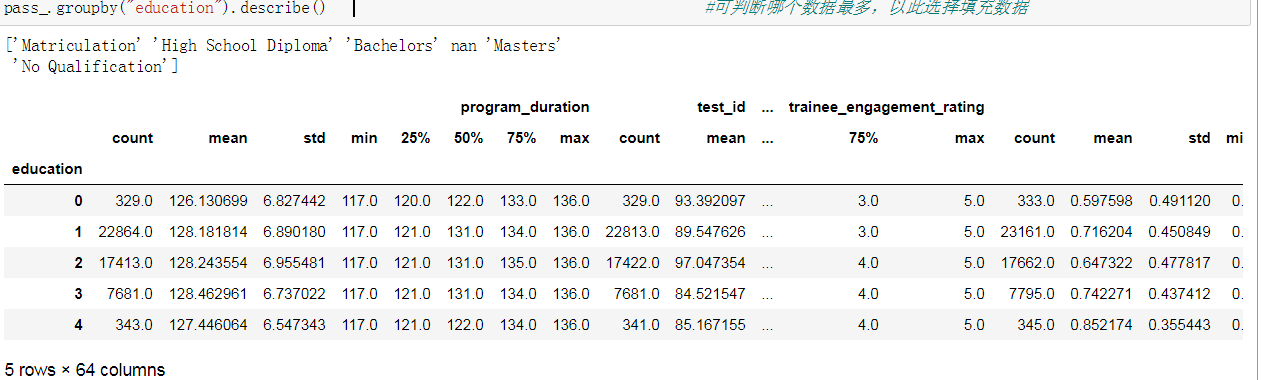
1 pass\_.head(10) 查看前十个数据，若不输入参数默认为前五个数据



2 pass\_.describe() 查看数据各个相关数值 平均数等

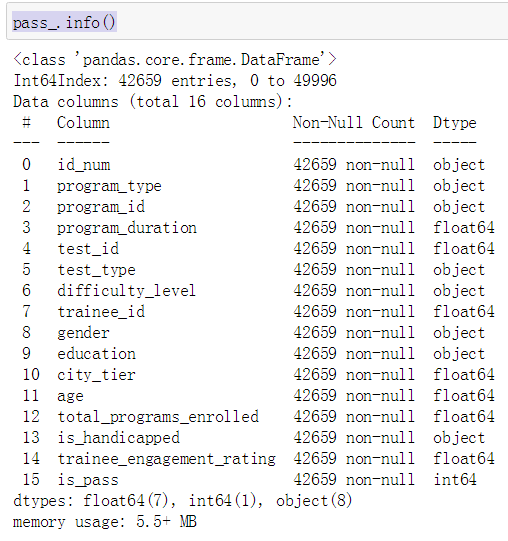


3 pass\_.groupby("education").describe() 查看"education"列的相关数值



可以用此来看此列数据详细分析

4 pass\_.info() 看看各列数据的总数



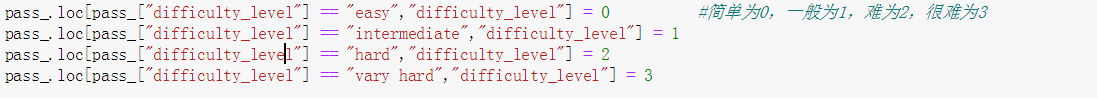
可以借此来看有无缺失值

3 数据预处理

此操作对测试集和训练集都要进行

1 pass\_.loc[pass\_["difficulty\_level"] == "vary hard","difficulty\_level"] = 3

把string型数据转化为特定的数字以便进行预测



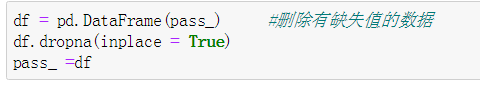
2 缺失值填充

 用F填充gender中缺失的数据



用中位数填充program\_duration的缺失值

3 有缺失值的行的删除



函数的拓展：



4 线性规划的实现

1 利用库实现线性规划

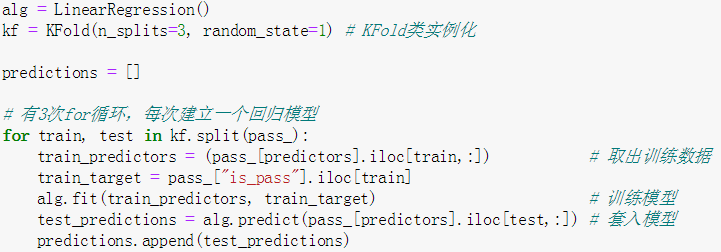
导入库



特征选择



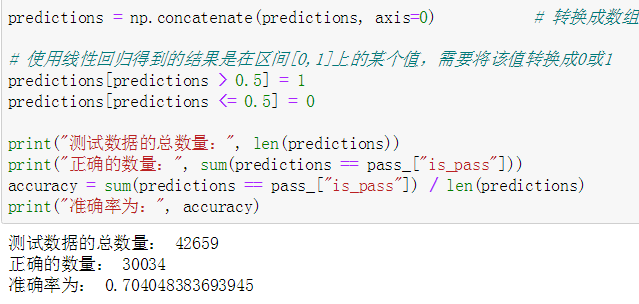
训练，预测数据



用了交叉验证法，把数据分为三份，每一份都训练相关方程



查看结果，准确率等



predictions[predictions > 0.5] = 1

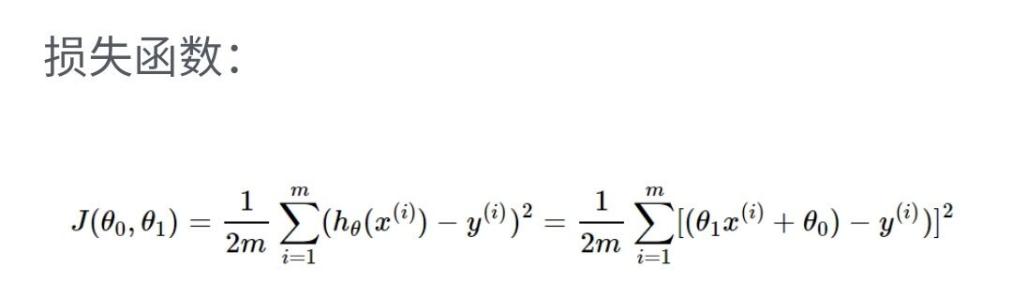
predictions[predictions <= 0.5] = 0

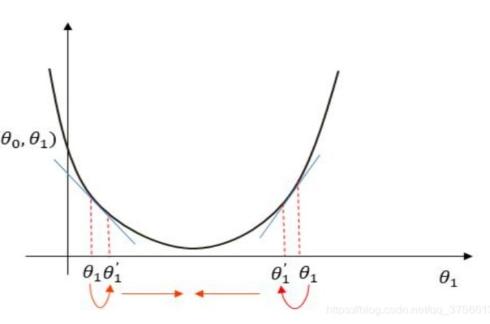
对回归的数据进行处理，使其变为0 , 1型

2 代码实现线性回归(梯度下降)

原理:

损失函数： 计算得到损失函数的最小值所对应的theta

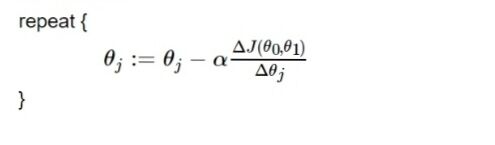




单元线性回归：

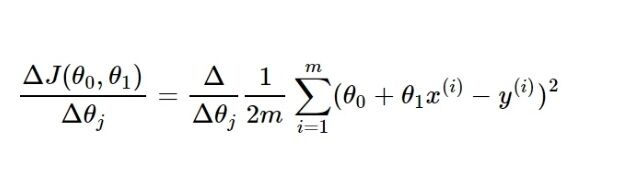
我们可以用梯度下降法来求得该点的参数值，每走一步都根据，所求的梯度来决定当前最优方向，然后继续前进，直到非常接近损失函数的最小值。

1. 初始化一个theta
2. 不断修改theta使损失函数越来越小

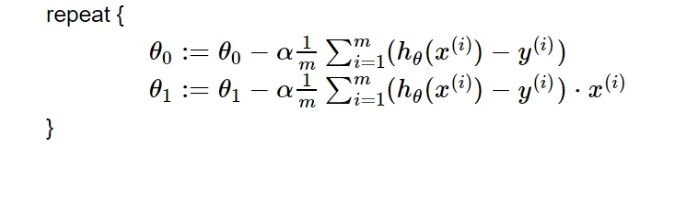


我们需要求解学习率a，以及梯度，而学习率是人为设定的，并不需要从训练集中学习，所以我们只要求解梯度即可

其公式为：

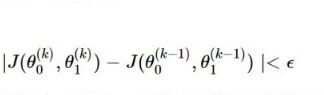


所以对于一元线性回归，梯度下降算法为

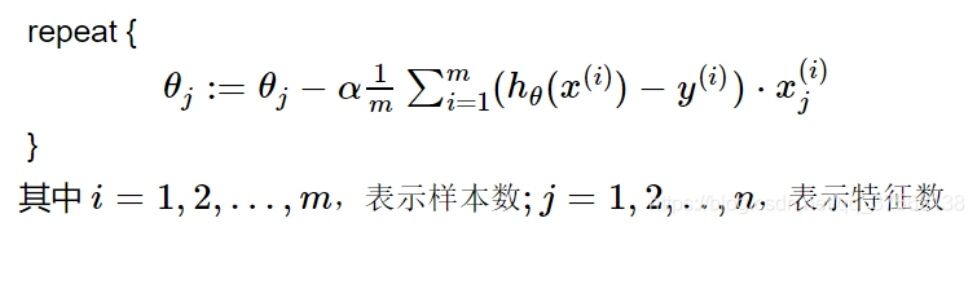


重复上述过程，直到收敛或达到最大迭代次数

判断条件为：

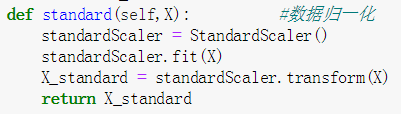


同理多元线性回归为：

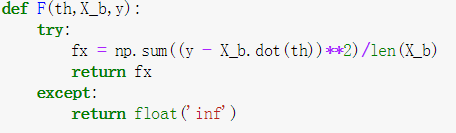


代码实现：

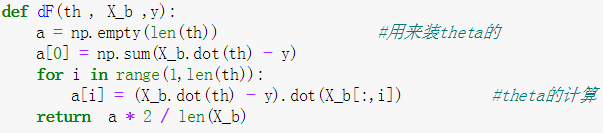
1 数据归一化 使模型更加拟合



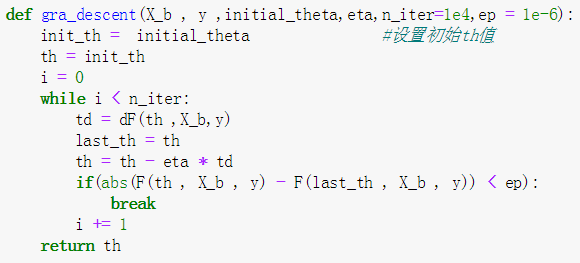
2 求损失函数的值



3 求此点对应的梯度

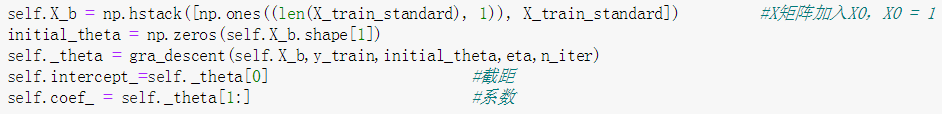


4 调用2,3来求出损失函数最小值所对应的theta

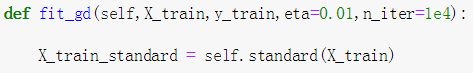


n\_iter为迭代的最大次数 ep是用于判断接近最小值的

5 调用4，得出线性函数的系数和截距



6 封装2,3,4,5，得到fit\_gd函数

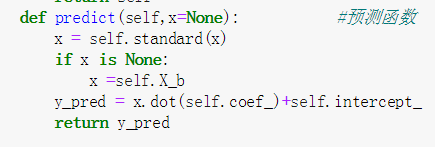


加2,3，4,5

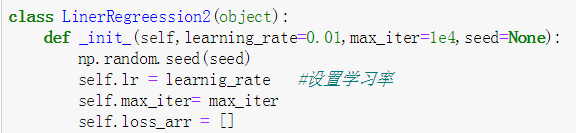


这是训练模型的函数

7 预测函数



8 加上初始化，封装此类



加6,7

5 逻辑回归的实现

1 调用库实现逻辑回归

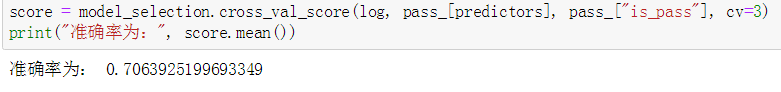
导入库



初始化



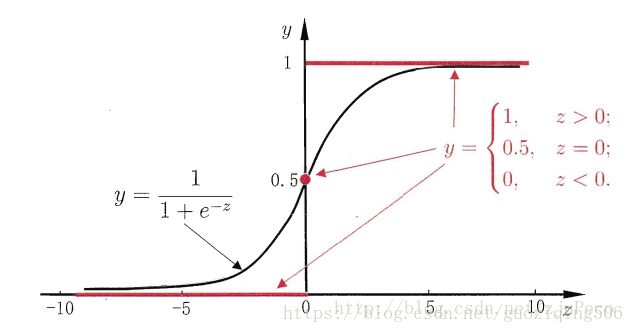
训练与预测与线性回归相同，直接看准确率：

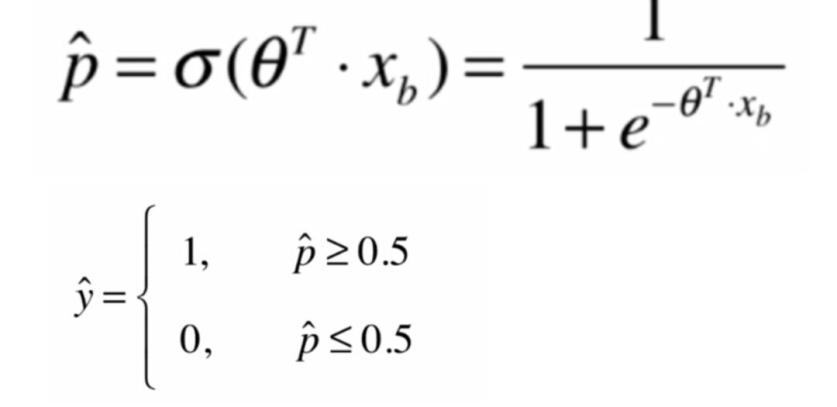


2 代码实现逻辑回归

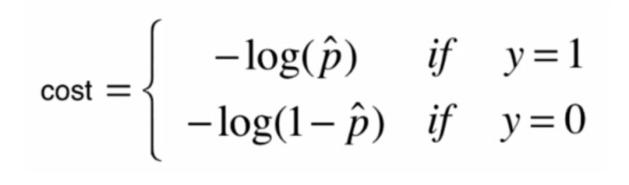
原理：

Sigmoid函数：

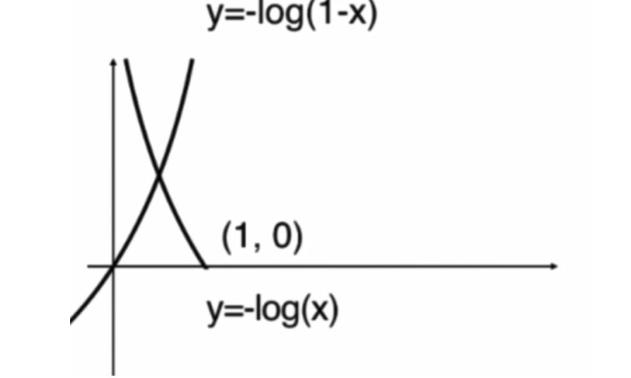




现在的问题就是,给定了一组样本数据集X和它对应的分类结果y，我们如何找到参数theta，使得用这样的方式可以最大程度的获得这个样本数据集X对应的分类输出y  
  
这就是我们在训练的过程中要做的主要任务,也就是拟合我们的训练样本,而拟合过程,就会涉及到逻辑回归的损失函数



图像为



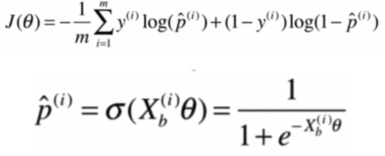
当y=1时，p趋近于零的时候，在这个时候可以看此时-log(p)趋近于正无穷，这是因为当p趋近于0的时候，按照我们之前的这个分类的方式，我们就会把这个样本分类成0这一类，但是这个样本实际是1这一类，显然我们分错了，那么此时我们对它进行惩罚，这个惩罚是正无穷的。

随着p逐渐的增高，可以看我们的损失越来越小，当我们的p到达1的时候，也就是根据我们的分类标准，我们会将这个样本x分类成1，此时，它和这个样本真实的y等于1是一致的，那么此时损失函数取0也就是没有任何损失。当y=0时同理。

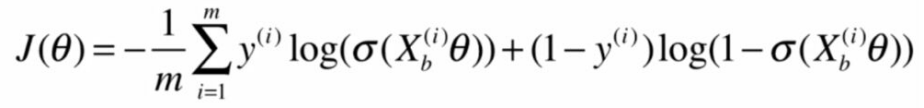
对损失函数进行简化得：



这样,根据我们求出的p,就可以得出这次估计的损失是多少  
最后,再把损失相加求平均值,其公式为:

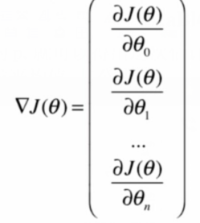


整合公式得

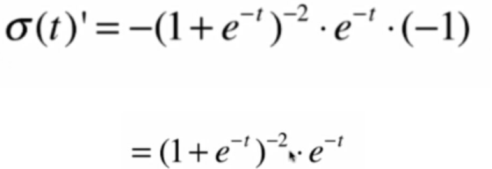


下面我们要做的事情,就是找到一组theta,使得J(theta)最小，我们可以使用梯度下降法求得它的解。

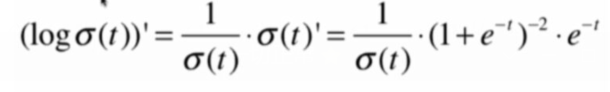
首先J(theta)的梯度公式：

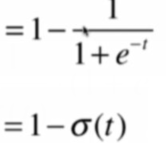


对sigmoid函数求导：

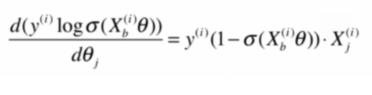


再对log函数求导：

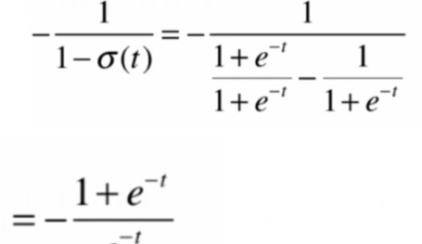




损失函数前半部分求导：

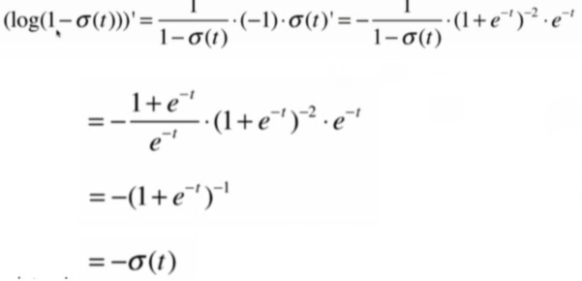


损失函数后半部分求导：

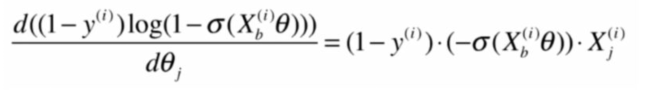




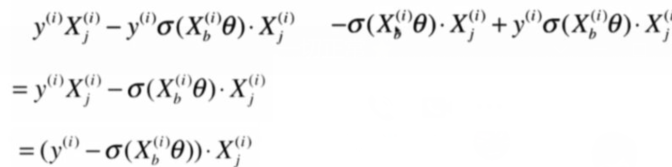
化简得：



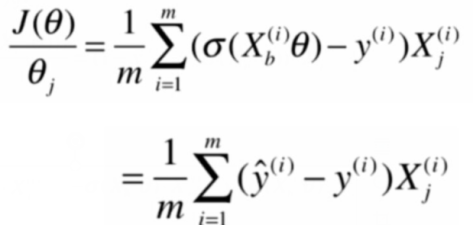
所以后半段推导结果为：



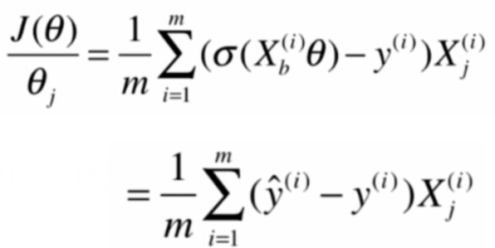
前后相加



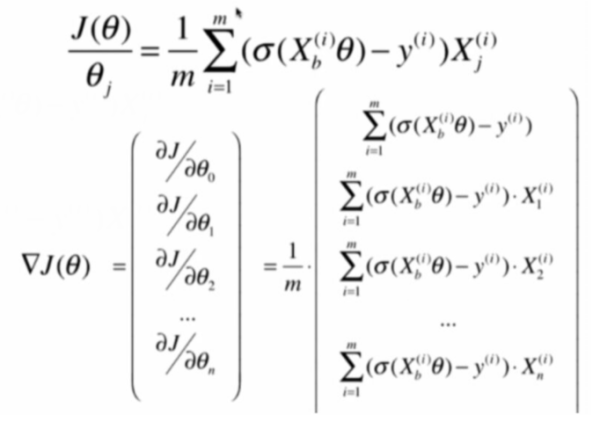
既



由此得到梯度



以及函数模型



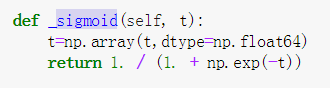
接下来使用梯度下降法即可

代码实现

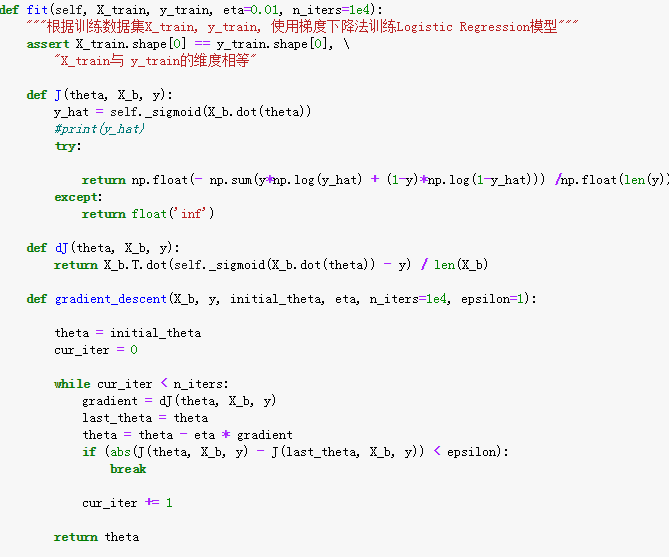
1 导入库



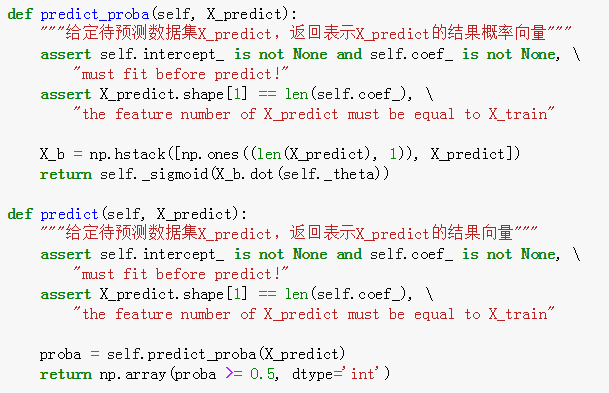
2 sigmoid函数



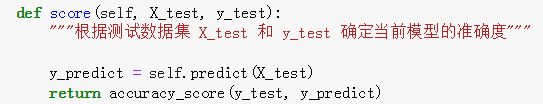
3 梯度下降法求解theta



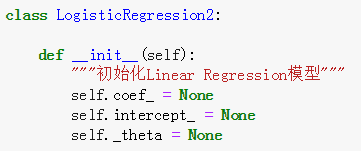
4 预测 先求结果概率向量，再求结果向量



5 准确度求解



6 封装函数

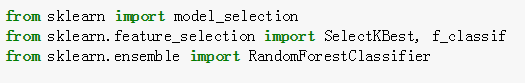


加2,3,4,5

6 随机森林模型

调用库实现随机森林

导入库



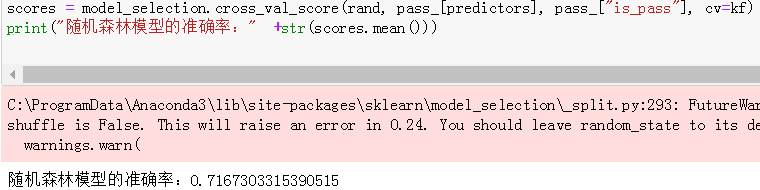
特征选择



初始化



准确率



查看各数据重要程度

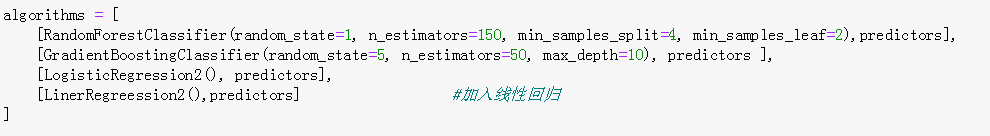


# 7 集成多个分类

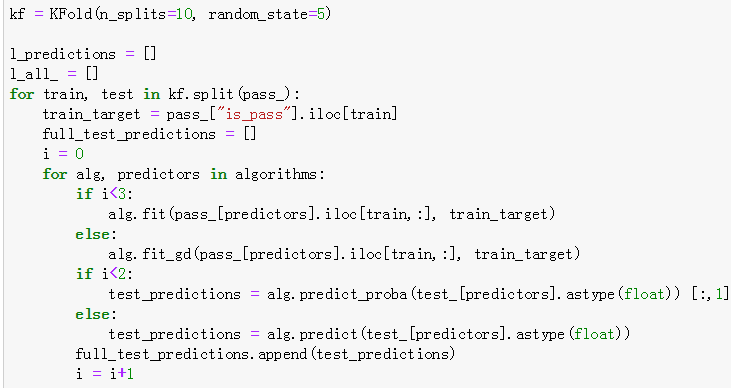
导入库



加入模型

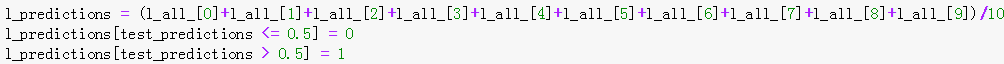


进行数据的训练及预测





此操作属于第一个for循环，将各个回归方程按比例进行预测，以得出更加优化的预测结果，然后将预测结果加入所设列表中



将交叉验证分的10次预测数据进行平均，再将所得数据进行处理得0,1型得到结果

8 将结果导入文件



所得文件第一列为id\_num，第二列为is\_pass即为结果