逻辑回归学习笔记

实验平台：Anaconda

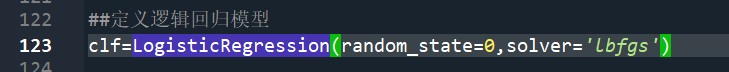


基础函数库中数学分析常用：numpy。例如：min,max,abs等函数。

数据分析常用pandas：特别是其中的DataFrame，可以将数据格式化，变成类似于表格的形式。一般采用字典的形式输入。



load了鸢尾花的数据后，可以利用data.dir()查看一下data里面的变量和属性名称。



LogisticRegression中自带正则化项，一般默认值为L2（还可选为L1）。在L2下，solver可选的算法{‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’}：

a) liblinear：使用了开源的liblinear库实现，内部使用了坐标轴下降法来迭代优化损失函数。

b) lbfgs：拟牛顿法的一种，利用损失函数二阶导数矩阵即海森矩阵来迭代优化损失函数。

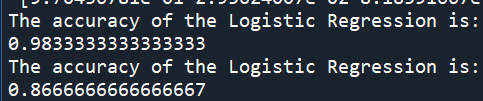
c) newton-cg：也是牛顿法家族的一种，利用损失函数二阶导数矩阵即海森矩阵来迭代优化损失函数。

d) sag：即随机平均梯度下降，是梯度下降法的变种，和普通梯度下降法的区别是每次迭代仅仅用一部分的样本来计算梯度，适合于样本数据多的时候，SAG是一种线性收敛算法，这个速度远比SGD快。

若将原来函数中的solver改变参数值，结果会有什么变化呢？

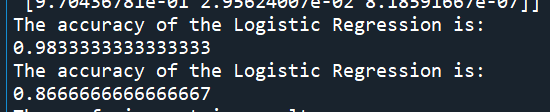
源程序结果：





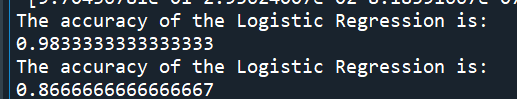
Solver=’lbfgs’:在训练集上的准确度已经接近100%，在测试集上也有很好的表现，准确度达到了大约87%。与原设置性能相同。





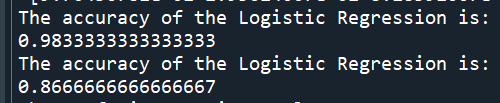
Solver=’newton-cg’:性能相同。



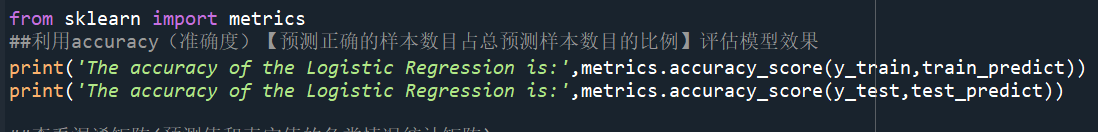


Solver=’sag’:性能相同。

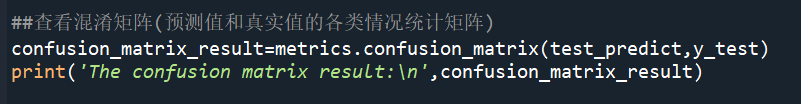


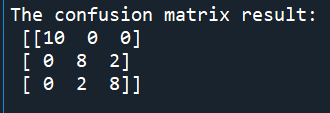


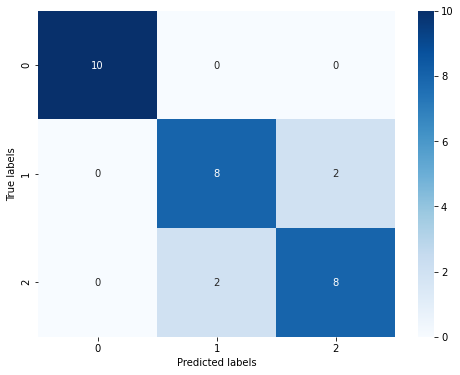
学习使用metrics库中的accuracy\_score函数来计算正确率：



查看混淆矩阵的时候最好借助图可视化来看，这样比较直观清楚！







可以看到在label=1的时候还是有错误的预测结果。

代码后思考：

需要正确认识逻辑回归中的梯度上升和梯度下降算法。利用sigmoid函数做梯度上升时一定要注意偏导值的正负的物理意义。梯度上升的基本思想就是找到函数的最大值，就沿着它的梯度的方向移动。梯度下降主要用来找到最小值。

疑问：梯度上升和梯度下降在逻辑回归中的区别？