

**模式识别大作业**

题 目 Criteo 展示广告点击率预测

学 院 信息科学与工程学院

专 业 信息与通信工程

组 员 秦祥翔

指导教师 赵海涛

**完成日期： 2018 年 10 月23日**

**模式识别作业报告——Criteo 展示广告点击率预测**

组员：秦祥翔

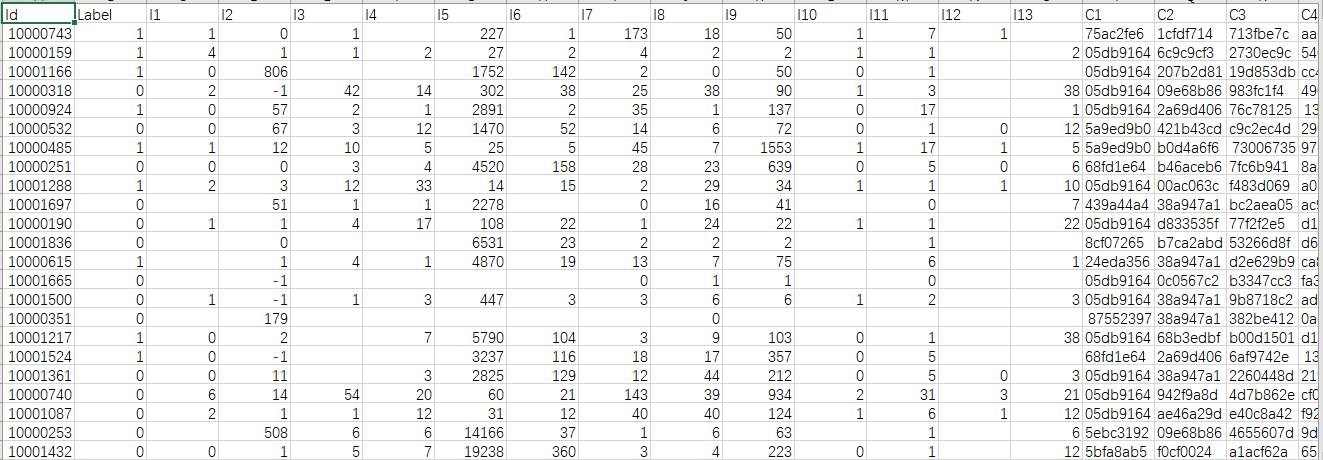
本学期选修了赵海涛老师的模式识别课程，受益匪浅。上周学习了Logistic回归的相关知识，对Logistic回归有了较深的了解，并通过本次针对Criteo 展示广告的数据集来预测用户是否会点击广告的实验来巩固所学的内容。

本次实验使用Regularized Logistic Regression模型，即正则化logistic回归，用来解决过拟合的问题，使用Logarithmic Loss作为最后评判标准，实验会根据测试集给出的用户访问记录，预测出用户点击广告的概率并存为CSV文件，上传到LintCode网站上得分为0.45599，暂时排名第九。

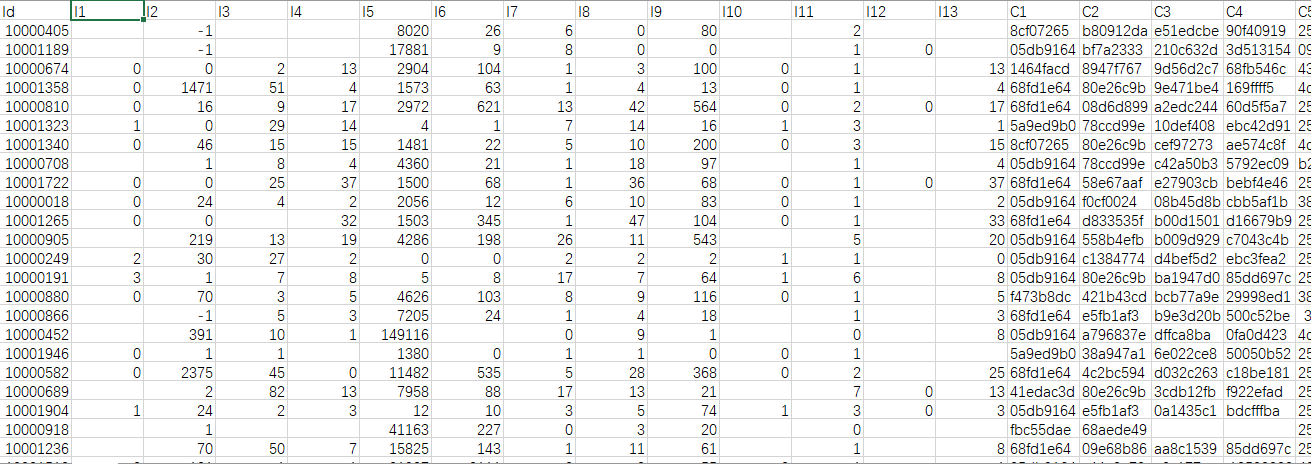
**一、数据集简介**

Criteo是一家第三方展示广告公司，与世界上超过4000家电子商务公司有合作关系。说到广告，关注的最多的就是点击率了。我们经常能听说某某科学家通过建立更好的点击率预测模型，为公司带来上亿的增量收入。本次实验我使用Criteo所共享的一周展示广告数据，数据中提炼了13个连续特征、26个离散特征和用户是否点击了该页面广告的标签，本次实验使用数据的前13个连续的数字特征，训练出合适的模型，预测用户在不同的特征下是否会点击广告。

本次实验主要使用的数据文件是train.csv（训练集）和test.csv（测试集）。其中数据文件train.csv提供了1599条的用户访问网页和点击广告记录的对应特征，l1～l13为计数特征，c1～c26为类别特征。Label表示用户是否点击广告，0为未点击，1为点击。如图1所示。数据文件test.csv与train.csv类似，提供了train.csv之后一段时间的用户访问网页和点击广告记录对应特征。如图2所示。



**图1. train.csv文件内容**



**图2. test.csv文件内容**

需要说明的是，两个数据集在特征上都存在一些缺失数据，即部分样本的某个特征值为nan，因此需要对数据进行缺值处理，本实验将每一维特征的均值对原数据进行缺值填充，然后进行标准化处理，使每一维特征的方差为1，均值为0。

可以通过某些算法对用户对某条广告或内容可能的点击率有一个预判，以便判断哪些条目应该被放在更重要的位置上。本次实验主要做的是预测用户是否会点击广告的概率，而后将预测的评分与submisson.csv中的真实结果进行对比，且通过计算两者的Logarithmic Loss（对数损失函数）值来评价预测的好坏。

**二、实验过程**

**2.1方案分析**

很明显，这是一个二分类的问题，二分类问题主流的方法有Logistic Regression，虽然叫Logistic回归，但是模型最终完成的是一个分类的任务。Logistic Regression的起源主要分为几个阶段，从开始想到logistic这个词，到发现Logistic Function，再推导出Logit Function，最后才命名Logistic Regression。Logistic起源于对人口数量增长情况的研究，作为Regression Analysis的一个分支，它实际上还受到很多Regression Analysis相关技术的启发。Logistic Regression是为了解决分类问题，根据一些已知的训练集训练好模型，再对新的数据进行预测属于哪个类。

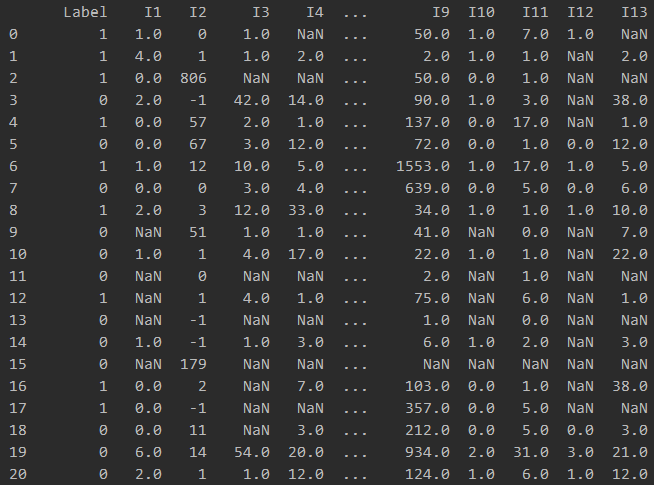
常用的分类模型还有GBDT（梯度提升决策树），又叫 MART（Multiple Additive Regression Tree)，是一种迭代的决策树算法，该算法由多棵决策树组成，所有树的结论累加起来做最终答案。它在被提出之初就和SVM一起被认为是泛化能力较强的算法。GBDT中的树是回归树（不是分类树），GBDT用来做回归预测，调整后也可以用于分类。GBDT的思想使其具有天然优势可以发现多种有区分性的特征以及特征组合。但无论用何种模型，本次实验的核心在于通过已有数据预测用户点击广告的概率。

鉴于上周刚刚学习了Logistic回归，为了加强对知识的理解并懂得如何应用，

我使用Regularized Logistic Regression模型来进行本次实验。正则化Logistic回归可以解决普通的Logistic回归过拟合的问题，其通过在损失函数中加入惩罚项来减小某些项的权重以防止对数据的过拟合。具体的算法思路会在下文进行详细的介绍。

**2.2数据预处理**

本次程序主要用Python编写，为进行数据训练及测试，必须读取对应的数据。读取数据主要使用了Pandas模块，可将数据集读取为数据框架，如图3所示：



**图3 原始数据的读入**

由于数据框架（Data Frame）不便于数据操作，我们需要对数据转换成矩阵，取其对应的特征列和标签列。对于测试数据的读取操作类似。将数据读取部分封装为函数，通过调用即可读取对应的数据。对于训练数据，没有对应的真实标签，且需要读取对应的ID以便模型进行预测。

同时，从图中可以看出某些特征存在缺失的情况，即图中红色标出的nan部分，这里需要将每一列特征的均值替代数据中缺失的部分。缺值处理的方式有很多种：

（1）忽略元组

当缺少类别标签时通常这样做（假定任务涉及分类时），除非元组有多个属性缺失值，否则该方法不是很有效。当每个属性缺少值的百分比变化很大时，它的性能特别差。

（2）人工填写缺失值

一般该方法很费时，并且当数据集很大，缺少很多值时，该方法可能行不通。

（3）使用一个全局常量填充缺失值

将缺失的属性值用同一个常数(如0)替换。如果缺失值都用0替换，则挖掘程序可能会认为它们形成一个有趣的概念，因为它们都具有相同的值。因此，虽然该方法很简单，但是它十分不可靠。

（4）使用与给定元组属同一类的所有样本的属性均值

如本次实验中的数据，使用具有相同信用度的给定元组的平均值替换原始数据中的缺失值。

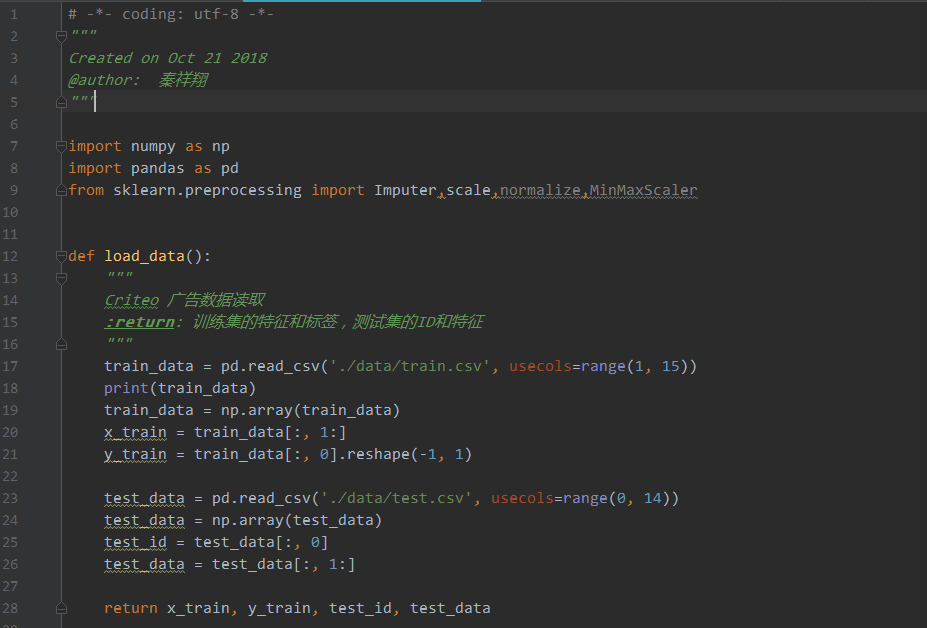
（5）使用最可能的值填充缺失值

可以用回归、使用贝叶斯形式化的基于推理的工具或决策树归纳确定。

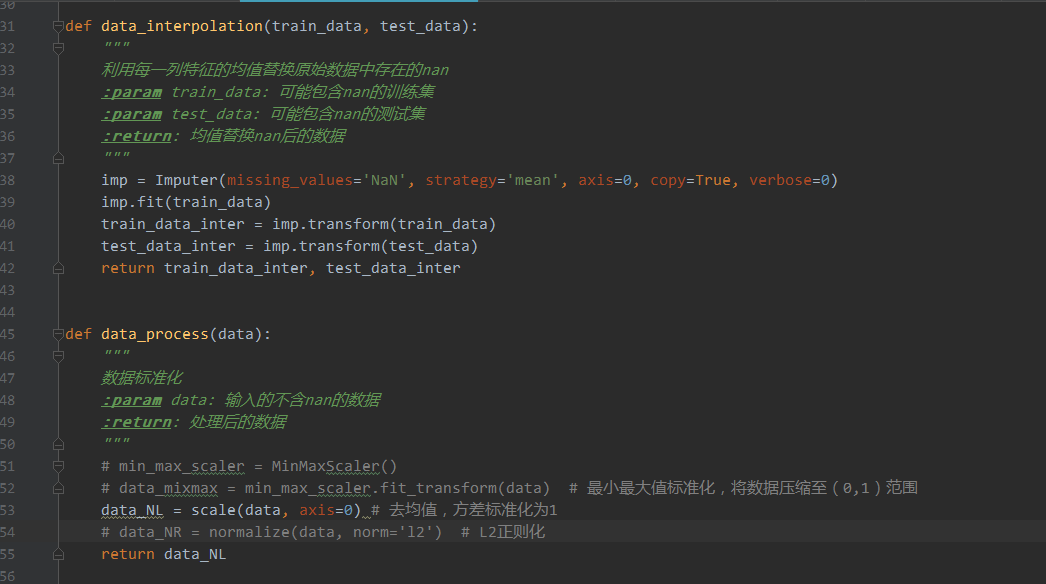
考虑到时间和算法的复杂问题，本次实验使用每一列特征的均值来填充每一维特征中所缺失的部分。调用sklearn.preprocessing.Imputer类进行缺值处理。

在进行缺值处理完以后，还需要对数据进行规范化处理，将数据的均值变为0，方差变为1。同时我还考虑了对数据进行标准化操作和将数据整体压缩到（0,1）之间，但分别进行这三种操作的实验结果差别细微，可以忽略，故可以只对数据进行规范化处理。

该部分Python程序如图4,5所示：



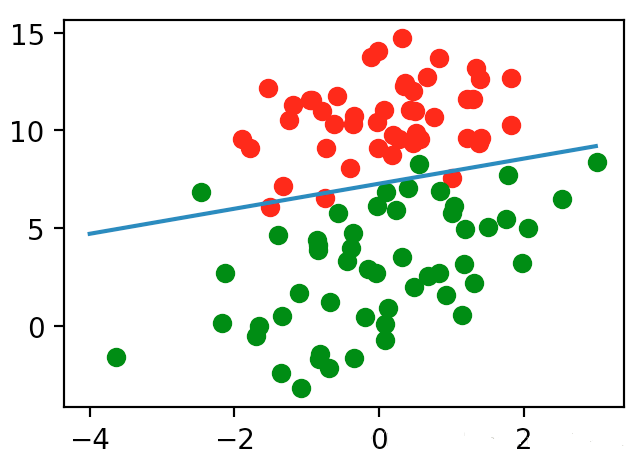
**图4 数据读入代码**



**图5 数据预处理代码**

**2.2算法原理及程序实现**

Logistic回归是为了解决分类问题，根据一些已知的训练集训练好模型，再对新的数据进行预测属于哪个类。如图6所示，也就是说逻辑回归的目标是找到一个有足够好区分度的决策边界，从而能够将两类很好的分开。假设已经存在这样一个边界，针对于图中这种线性可分的情况，这条边界是输入特征向量的线性组合。



**图6 逻辑回归图例**

假设输入的特征向量为，*Y*的取值只有两种：0和1。那么决策边界可以表示为，假如存在一个样本使得，那么可以将其判别为一类，反之为另一类。这个过程实际上是感知机，即只通过决策函数的符号来判断属于哪一类。而Logistic回归需要再进一步，它要找到分类概率*P*(*Y=*1)与输入向量*x*的直接关系，然后通过比较概率值来判断类别，sigmoid函数便可以满足这样的一个要求，其公式为：

（2-1）

sigmoid函数可以将输入映射到（0，1）范围内，这可以让得到的通过sigmoid函数得到一个介于0和1的概率值，假设二分类问题满足伯努利分布，则有:

（2-2）

（2-3）

其中称为权重，称为偏置，其中的看成对的线性函数。然后对比上面两个概率值，概率值大的就是x对应的类。有时候为了书写方便，会将写入，即，其中 ，这样，可以将 写成 ，其中，其中，为 的n维特征。并把 写作 。

由式（2-2）和式（2-3）可以得到我们的后验概率：

（2-4）

有了上面的分类概率，就可以建立似然函数，通过极大似然估计法来确定模型的参数,对于有m个样本的数据以及每个样本所对应的标签 ，假设样本之间是相互独立的，即似然函数为：

（2-5）

对上式两边取对数得：

（2-6）

算法的目标即最大化这个似然函数，优化的方法有很多，梯度下降，牛顿法和BFGS等等，本实验使用的是梯度下降法进行优化。将式（2-6）改写为logistic回归的损失函数：

（2-7）

对式（2-7）求对的偏导：这里只考虑一个实例产生的似然函数对一个参数的偏导。

（2-8）

梯度下降是通过J(w)对w的一阶导数来找下降方向，并且以迭代的方式来更新参数，更新方式为

（2-9）

其中，为迭代次数，为学习率，每次更新参数后，可以通过比较或者与某个阈值大小的方式来停止迭代，或者自行指定迭代次数。

值得一提的是，本次实验使用了正则logistic回归。logistic回归通过正则化（regularization）惩罚参数，防止其取得过大，可以避免过拟合问题（overfitting），即在损失函数中加入惩罚项：

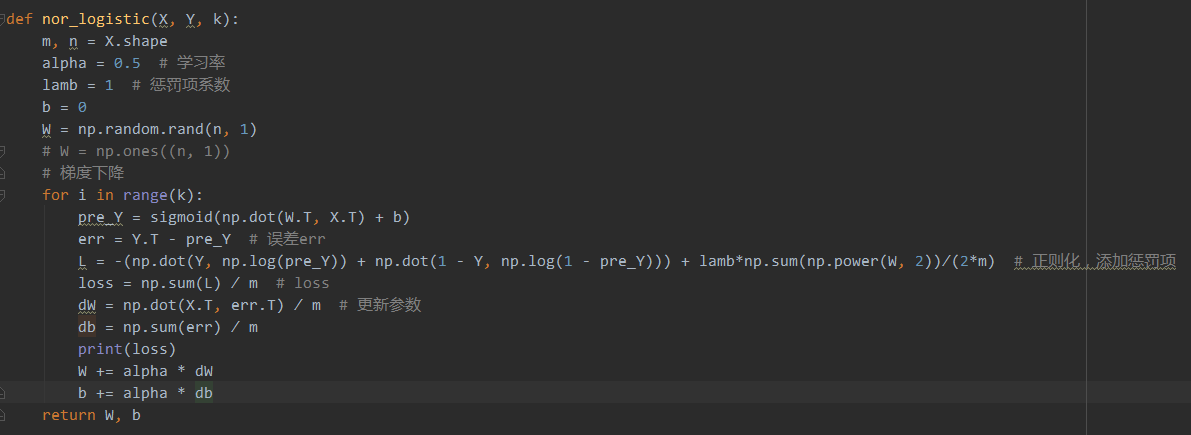
（2-10）

其中，为惩罚项系数，该惩罚中是不包含项的，即对进行不同的更新方式：

（2-11）

（2-12）

该部分的核心算法的Python程序实现代码如图7所示：



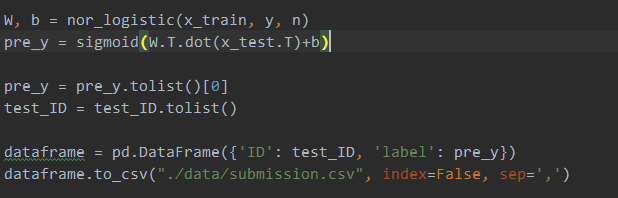
**图7 正则化logistic回归代码**

**2.3结果评价**

如前文所述，本次实验以计算Logarithmic Loss来评判预测的准确度，Logarithmic Loss越小则说明评分越正确。计算公式如下：

（2-13）

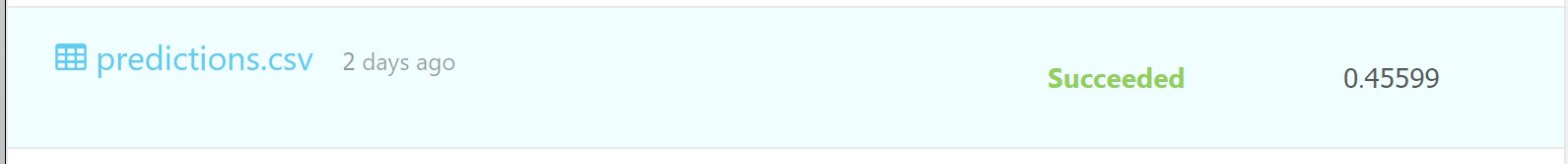
其中，为样本数，为类别数，本次实验，是真实的类别标签，本次实验为0或者1, 0表示用户未点击广告，1表示用户点击了广告。本实验只需要将预测的结果存成submission.csv文件上传到网站上进行结果评价。代码如图8所示。



**图8 结果评价代码**

**2.4 预测结果**

将结果文件上传到LintCode网站上得到的分数为0.45599，排名第九。如图9所示。





**图9 预测结果**

**三、小组分工**

程序设计及编写：秦祥翔

程序调试：秦祥翔

实验报告：秦祥翔

**四、作业总结**

因我在本科期间接触过相关的知识，所以在算法的程序实现上没有遇到太多的阻碍，在数据预处理上，对于缺失的特征值，尝试过很多不同的方案，但效果大同小异，故使用均值填充对应列的缺值，数据归一化过程也尝试了不同的方法，但由于本人能力有限，并没有调试出更好的方案。所以最终结合结果和计算时间考虑，以此目前方法较优。这次的大作业让增加了我对模式识别的兴趣，同时原来神秘的分类算法变得立体清晰。感谢赵老师教学帮助，并且在作业过程中给了我们大家很多指导。