

**模式识别大作业**

题 目 Criteo 展示广告点击预测

学 院 信息科学与工程

专 业 信息与通信工程

组 员 付锡欣

指导教师 赵海涛

**完成日期： 2018 年 10 月24日**

**模式识别作业报告——Criteo 展示广告点击预测**

组员：付锡欣

通过本学期模式识别课程的学习，在赵海涛老师的辛勤指导下，我对模式识别中的Logic Regression有了一定了解，并通过本次针对Criteo广告点击的预测实验进行进一步的巩固。经过一段时间的程序编写以及代码调试，最终完成了Criteo广告点击的预测的实验。接下来，我详尽地说明此次实验过程。

**一、Criteo广告点击数据集说明**

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。本题我们使用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，如表1所示。

**表1 数据集示例**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Label | I1 | I2 | I3 | I4 | I5 | I6 | I7 | I8 | I9 |
| I10 | I11 | I12 | I13 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 | C16 | C17 | C18 |
| C18 | C19 | C20 | C21 | C22 | C23 | C24 | C25 | C26 |  |  |

提供的所有数据集均为csv文件。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)，最终结果保存为result.csv。第一列为用户ID(唯一性)；第二列为真实输出结果Label,即是否点击广告；第三列至第十五列为13个连续特征I1-I13；第十六列至第四十一列为26个离散特征。

本次实验主要是，根据train.csv进行广告点击率预测模型的训练，训练完成后，对test.csv中的用户根据所提供的特征值进行点击率的预测。最终，根据Logarithmic Loss作为最后评判标准，与submission.csv进行对比，来评价此次预测模型的好坏。

**二、整体解决方案**

**2.1方案分析**

(logistic regression)属于概率型非线性回归，它是研究二分类观察结果与一些影响因素之间关系的一种多变量分析方法。在流行病学研究中，经常需要分析疾病与各危险因素之间的定量关系，为了正确说明这种关系，需要排除一些混杂因素的影响。传统上常常使用Mantel-Haenszel分层分析方法，但这一方法适用于样本含量大，分析因素较少的情况。如果用线性回归分析，由于应变量Y是一个二值变量(通常取值1或0)，不满足应用条件，尤其当各因素都处于低水平或高水平时，预测值Y值可能超出0~1范围，出现不合理都现象。用logistic回归分析则可以较好的解决上述问题。

在社会科学诸如社会学、心理学、人口学、政治学、经济学以及公共卫生学当中，大量的观测因变量是二分类测量，在分析二分类因变量时最常使用的统计分析模型之一Logistic回归模型。

二分类通常有以下几种方法：

1. 贝叶斯分类法: 它通过预测一个给定的元组属于一个特定类的概率，来进行分类。朴素贝叶斯分类法假定一个属性值在给定类的影响独立于其他属性的 —— 类条件独立性。
2. 决策树：是一种简单但广泛使用的分类器，它通过训练数据构建决策树，对未知的数据进行分类。决策树的每个内部节点表示在一个属性上的测试，每个分枝代表该测试的一个输出，而每个树叶结点存放着一个类标号。
3. 支持向量机：把分类问题转化为寻找分类平面的问题，并通过最大化分类边界点距离分类平面的距离来实现分类。

在模式识别课程的学习中，我了解到了logistic 回归，对其他模型不再进行赘述。在此次实验中，我利用logistic模型进行广告点击率的预测。具体实现算法在下文中进行详细地解释。

**2.2数据结构分析**

根据此数据，我们选用了该数据中13个连续特征，用于点击率模型的训练，示例数据如表1所示。该数据为csv文件，第一列为用户ID(唯一性)；第二列为真实输出结果Label,即是否点击广告；第三列至第十五列为输入的13维特征I1-I13。

**表2 所用训练数据示例**



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)，最终结果保存为result.csv。特别说明，从表1可以看出，由于训练数据中部分特征值缺失，所以此次实验必须进行数据预处理。

第一列数据表示的是用户ID，不同ID代表不同用户 ，具有唯一性；

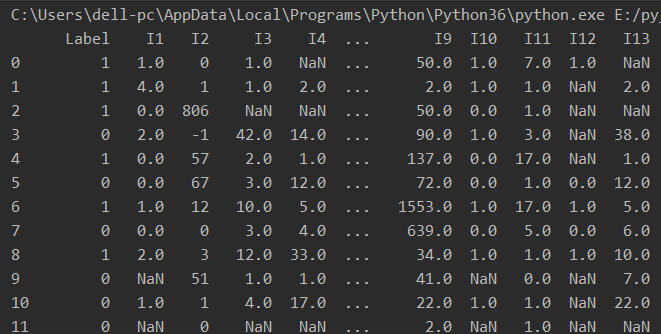
第二列数据表示Label，表示训练集中各个用户是否点击过广告，是训练广告点击率预测的真实结果输出；

第三列至第十五列数据代表每个用户所对应的十三维特征值，这是数据集的关键信息，是训练点击率预测的输入数据；

最终，考虑到时间以及编程复杂度问题，在原始数据表中我们只采用上述十五维数据来进行广告点击率预测模型的训练。

**2.2数据读入**

本次程序主要用Python编写，为进行数据训练及测试，对csv数据进行读取。本次实验读取数据使用了Pandas模块，由于只使用数据集的前十五列，所以将数据集的前十五列读取为数据框架，如图1所示：



**图1 原始数据的读入**

由于数据框架（Data Frame）不便于数据操作，可将其转为数组形式(array)。将输入特征值和标签分别存于x，y两个数组中。对于测试数据的读取操作，与训练集相似，但需要存储ID。

该部分Python程序如下所示：

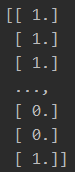
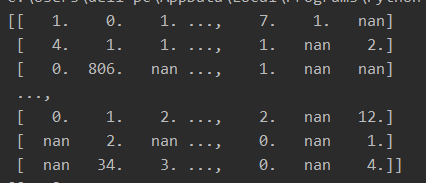
代码如下：

|  |
| --- |
| # =======================================  # 数据读入  d = pd.read\_csv(r".\data\train.csv",usecols=range(1, 15)) print(d) d = np.array(d) # print(d.shape) x = d[:, 1:] y\_train = d[:, 0].reshape(-1, 1)  # =======================================  d = pd.read\_csv(r".\data\test.csv", usecols=range(0, 14)) d = np.array(d) x = d[:, 1:] Id = d[:, 0] |

**2.3数据预处理**

**2.3.1缺值填充**

对训练数据进行观察，如图2所示，训练集中特征值存在缺值现象，需要进行数据预处理。缺值处理通常有五种处理方法：忽略元组、人工填写缺失值、使用一个全局常量填充缺失值、使用与给定元组属同一类的所有样本的属性均值、使用最可能的值填充缺失值。在此次试验中，我采用的是第四种方法：使用与给定元组属同一类的所有样本的属性均值。



**图2 训练数据的x，y**

该部分Python程序如下所示：

代码如下：

|  |
| --- |
| # =======================================  # 缺值处理添加均值  from sklearn.preprocessing import StandardScaler, Imputer import numpy as np  def insert\_nan(data):  imp = Imputer(missing\_values='NaN', strategy='mean', axis=0)  imp.fit(data)  return imp.transform(data)  # ======================================= |

**2.3.2数据标准化**

数据的标准化是将数据按比例缩放，使之落入一个小的特定区间，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。标准化的优点主要是：(1)加快梯度下降求最优解的速度；(2)有可能提高精度。

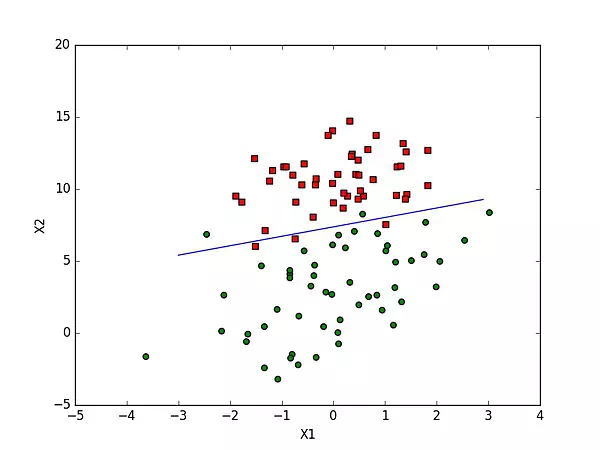
该部分Python程序如下所示：

代码如下：

|  |
| --- |
| # =======================================  # 数据标准化  def NL(data):   scaler = StandardScaler()  scaler.fit(data)  scaler.transform(data)  data = np.mat(data)  return data  # ======================================= |

**2.4算法原理及程序实现**

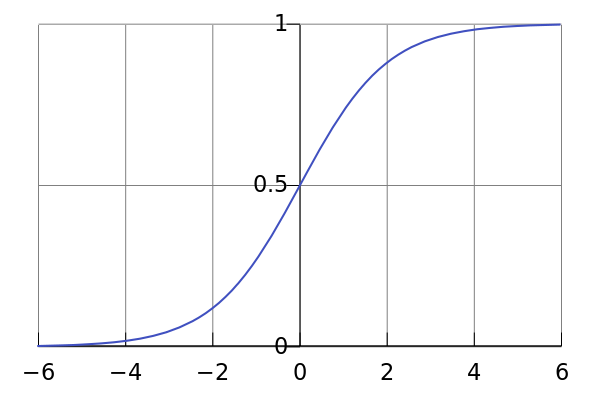
Logistic回归用于解决二分类问题，根据已知的训练集进行模型训练，再用此模型对新的数据进行分类预测，如图3所示。



**图3 基于logistic回归分类示例**

回归问题常见步骤是：寻找g函数；构造J函数；想办法使得J函数最小并求得回归参数。logistic回归的g(x)函数为：

（2-1）



**图4 logistic函数图像**

其中，g(x)表示取1的概率值。图像如图4所示。

那么对于输入x的分类结果对于类别1和类别0的概率分别为：

（2-2）

（2-3）

那么对于构造损失函数J，它们基于最大似然估计推到得到的：

（2-4）

取似然函数：

（2-5）

通过梯度下降求解梯度最小值：

（2-6）

其中，k为迭代次数，α为学习率。

代码如下：

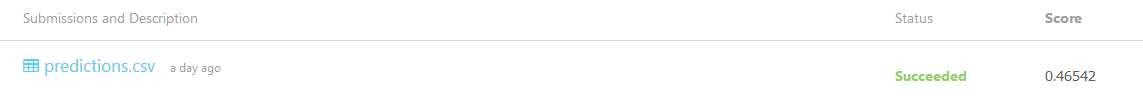
|  |
| --- |
| def sigmoid(Z):  return 1.0 / (1 + np.exp(-Z))  def gradA(X, Y, k):  m, n = X.shape  alpha = - 0.5  b = 0  W = np.ones((n, 1))  # 梯度下降  for i in range(k):  Z = np.dot(W.T, X.T) + b  A = sigmoid(Z)  dZ = A - Y.T # 误差err  dW = np.dot(X.T, dZ.T) / m # 更新参数  db = np.sum(dZ) / m  L = -(np.dot(Y, np.log(A)) + np.dot(1 - Y, np.log(1 - A)))  J = np.sum(L) / m # loss  print(i, J)  W = W + alpha \* dW  b = b + alpha \* db  return W, b |

**2.4评定测试**

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：

（2-7）

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，预测的点击率越靠近0/1，代表用户极可能未点击广告(点击广告)。本实验需要将预测的结果存成result.csv文件上传到lintcode上，进行分数评估。如图5所示，最终分数为0.46542。

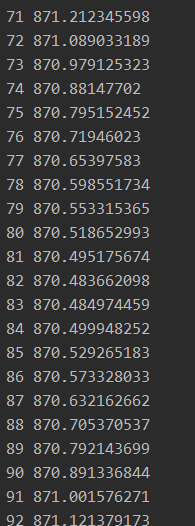


**图5 最终评估结果**

**2.5调试及预测结果**

在最终的程序调试中，迭代次数为80左右时，得到最优解L=870。

图6为实验调试时所得最优解过程：



**图6 运行结果（参数：K=200， alpha=-0.5）**

**三、小组分工**

程序设计及编写：付锡欣

程序调试：付锡欣

实验报告：付锡欣

**四、作业总结**

这是刚开始接触模式识别，算法的代码写起来很生疏，在算法实现上出现了很多错误。由于一开始没有进行数据缺值处理，导致数据虽然成功导入，但是没有办法进行分类。之后，通过查看数据发现了该问题，通过百度的方式查找到了数据缺值的处理方法并成功改正。数据缺值处理后，直接利用logistic回归进行点击率预测，结果不是很理想。在同学的提醒下，对数据进行标准化处理，使其均值为0，方差为1，再次进行点击率预测。此次结果与之前结果相比有了很大的提升。

在编写代码的过程中，我发现自己有很多代码编写不规范，希望通过以后的联系，逐渐改正，规范自己的代码。这次大作业使我将理论应用于实践，明白了logistic回归的作用，使得原本不够理论的东西变的更加有用。感谢赵老师以及同学的帮助，让我顺利地完成了此次大作业，而且有很大的收获。

**附：文件说明**

本次附件一共包含有：

1 大作业报告；

2 最终的Python实现程序源码：logist\_criteo.py、data\_processing.py

3 导出了预测数据集：result.csv