特征实验

**问题描述**

我们提供了在双11促销期间一组商人他们相应增长的买家。你的任务是预测未来这些新买家是否会成为这些商家的忠诚客户。换句话说，您需要预测这些新购买者在6个月内再次从同一商家购买商品的可能性。

**数据描述**

数据集包含匿名用户在“双11”之前和过去6个月的购物日志，以及他们是否是重复购买者的标签信息。由于隐私问题，对数据的采样存在偏差，因此该数据集的统计结果会偏离实际情况，但不会影响解的适用性。

**数据预处理**

1. 数据清洗
2. 检测异常样本
3. 缺省值处理：缺省值极多，非连续特征缺省值适中，连续特征缺省值适中，缺省较少
4. 数据采样

**（1）从负样本中抽取部分样本出来和正样本结合**

**（2）正样本重复若干次**

**（3）代价敏感学习**

**（4）SMOTE即合成少数类过采样技术**

**实验代码**

**import** numpy **as** np  
**import** xlrd *#引入xlrd模块***import** xlwt *#引入xlwd模块***def** open\_excel(file):  
 **try**:  
 data = xlrd.open\_workbook(file) *#打开excel文件* **return** data  
 **except** Exception **as** e:  
 print(str(e))  
  
**def** split\_age\_range(age):  
 *"""  
 将特征值年龄进行离散化为8个特征值* **:param** *age: 年龄区间值* **:return***: 离散化后的特征  
 """* **if** age == 0:  
 **return** [1,0,0,0,0,0,0,0,0]  
 **elif** age == 1:  
 **return** [0,1,0,0,0,0,0,0,0]  
 **elif** age == 2:  
 **return** [0,0,1,0,0,0,0,0,0]  
 **elif** age == 3:  
 **return** [0,0,0,1,0,0,0,0,0]  
 **elif** age == 4:  
 **return** [0,0,0,0,1,0,0,0,0]  
 **elif** age == 5:  
 **return** [0,0,0,0,0,1,0,0,0]  
 **elif** age == 6:  
 **return** [0,0,0,0,0,0,1,0,0]  
 **elif** age == 7 **or** age == 8:  
 **return** [0,0,0,0,0,0,0,1,0]  
 *# elif age == 8:  
 # return [0,0,0,0,0,0,0,0,1]***def** split\_gender(gender):  
 *"""  
 将特征值性别进行离散化* **:param** *gender:* **:return***: 返回离散化的特征  
 """* **if** gender == 0:  
 **return** [1,0,0]  
 **elif** gender == 1:  
 **return** [0,1,0]  
 **elif** gender == 2:  
 **return** [0,0,1]  
  
**def** split\_log(Log):  
 *"""  
 分割数据文件中的Log数据* **:param** *Log: Log数据* **:return***: 处理后的特征值  
 """* items = Log.strip().split(**'#'**)  
 purchase = 0;total = 0  
 click = 0;add\_to\_card = 0;add\_to\_favourite = 0  
 **for** i **in** range(len(items)):  
 total += 1  
 item = items[i].strip().split(**':'**)  
 **if** item[4] == **'2'**:  
 purchase += 1  
 **if** item[4] == **'1'**:  
 add\_to\_card += 1  
 **if** item[4] == **'3'**:  
 add\_to\_favourite += 1  
 **return** [float(total),float(round(purchase/total,3)),float(add\_to\_card),float(add\_to\_favourite)]  
  
**def** loadDataSet(path, training\_sample,colnameindex=0,by\_name=**u'Sheet1'**):  
 *"""  
 加载数据* **:param** *path: 数据文件存放路径* **:param** *training\_sample: 数据文件名* **:param** *colnameindex: 文件列名下标* **:param** *by\_name: 表名* **:return***: 数据集和类别标签  
 """* dataMat = [];  
 labelMat = [] *# 定义列表* filename = path + training\_sample  
 data = open\_excel(filename) *#获取文件* table = data.sheet\_by\_name(by\_name) *# 获取Sheet1* nrows = table.nrows *# 拿到总共行数* colnames = table.row\_values(colnameindex) *# 某一行数据 ['user\_id', 'age\_range', 'gender', 'merchant\_id','label']* **for** rownum **in** range(1, nrows): *# 也就是从Excel第二行开始，第一行表头不算* row = table.row\_values(rownum)  
 **if** row[1] == **'' or** row[2] == **'' or** row[5] == **''**:  
 **continue  
 if** row:  
 app = [] *#定义列表* app = split\_age\_range(row[1])+split\_gender(row[2]) + split\_log(row[5]) *# 将Log转化为特征值* dataMat.append(app)   
 labelMat.append(float(row[4])) *# 获取类别标签* **return** dataMat, labelMat  
  
**def** main():  
 *"""  
 主函数* **:return***: null  
 """* wb = xlwt.Workbook()  
 ws = wb.add\_sheet(**'sheet1'**,cell\_overwrite\_ok=**True**)  
 path = **"D:"** training\_sample = **'feature.xlsx'** *# 训练数据文件* trainingSet, trainingLabels = loadDataSet(path, training\_sample) *# 取训练数据  
 # print(len(trainingSet))* num = len(trainingSet)  
 **for** i **in** range(num):  
 **for** j **in** range(16):  
 ws.write(i,j,trainingSet[i][j])  
 ws.write(i,j+1,trainingLabels[i])  
 wb.save(**'D:\\featuredata.xls'**)  
 print(**"处理完成"**)  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 **"""  
 程序入口  
 """** main()

·在实验中引入xlrd，xlwt，sklearn两个模块

其中xlrd是读excel，xlwt是写excel的库

Sklearn是进行标准化数据需要引入的

使用xlrd，xlwt读写xls文件以及基本用法

·将特征值年龄离散化并转换为特征值

·将特征值性别离散化

·分割数据文件的Log数据

·加载数据

·data=xlrd.op r   
table=data.sheets()[0]          #通过索引顺序获取  
table=data.sheet\_by\_index(sheet\_indx)) #通过索引顺序获取  
table=data.sheet\_by\_name(sheet\_name) #通过名称获取

·获取整行和整列的值（数组）

table.row\_values(i)

table.col\_values(i)

获取行数和列数

nrows = table.nrows

ncols = table.ncols