**通过移动设备行为数据预测使用者的性别和年龄**

TalkingData Mobile User Demographics

Author： Zhiwen Gu

*此项目为入门项目，用来了解：*

*1.多表连接，数据处理；*

*2. OneHot编码*

*3. 特征选择；*

*4. 交叉验证选择参数*

**项目介绍：**

通过行为习惯对移动用户人口属性（年龄+性别）进行预测

• 数据及包含~20万用户数据，分成12组，同时提供了用户行为属性，如：手机品牌、型号、 APP的类型等

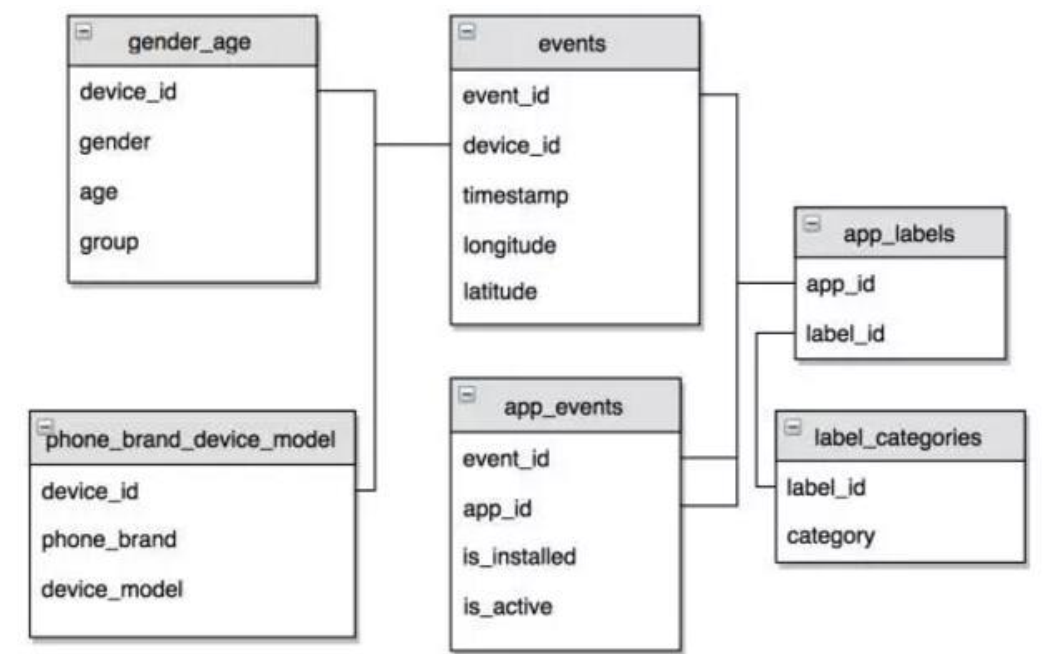
• 步骤：

1. 解读数据

2. 特征工程

3. 模型调参

• 数据集结构



• 数据集说明：

每个用户用一个ID表示，一个用户的行为是在一系列的Events里面，每个Event里面的信息包括该ID行为发生的时间、地理坐标信息，安装的APP类型、手机型号类别等

**数据集基本信息：**

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 74645 entries, 0 to 74644

Data columns (total 4 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 device\_id 74645 non-null int64

1 gender 74645 non-null object

2 age 74645 non-null int64

3 group 74645 non-null object

dtypes: int64(2), object(2)

memory usage: 2.3+ MB

**特征内容:**

device\_id： 设备id (主键)

gender：用户性别

age：用户年龄

event\_id：事件id标识

group：用户分组

phone\_brand: 手机品牌

device\_model: 设备型号

timestamp: 事件发生时间

……

**项目流程：**

**1. 分割数据集**

将原数据划分为训练集和测试集，用来验证训练出的模型准确率

**2. 加载数据**

index\_col='device\_id'；index\_col='event\_id'

加载时，设定各表之间的关联键，实现表的连接

**3. 特征工程**  
3.1 手机品牌特征

* 使用LabelEncoder将类别转换为数字，
* 使用OneHotEncoder将数字转换为OneHot码，稀疏矩阵

3.2 手机型号特征

* 合并手机品牌与型号字符串产生新的特征继续使用OneHotEncoder编码

3.3 安装app特征

* 将表“event”和“app\_events“进行连接，获取运行app的总次数和运行app的个数

n\_run\_s=device\_app['app\_id'].groupby(device\_app['device\_id']).size()

n\_app\_s=device\_app['app\_id'].groupby(device\_app['device\_id']).nunique()

* 填充缺失数据，没有的数据简单粗暴，直接填0

3.4 合并所有特征

tr\_feat=np.hstack((tr\_brand\_feat.toarray(),tr\_model\_feat.toarray(),tr\_app\_feat,tr\_run\_feat))

te\_feat = np.hstack((te\_brand\_feat.toarray(),te\_model\_feat.toarray(), te\_app\_feat, te\_run\_feat))

合并后：每个样本特征维度： 1800

tr\_app\_feat.shape: (1866, 1800) te\_app\_feat.shape: (1866, 1800)

3.5 特征范围归一化，这里选用标准归一化

3.6 特征选择，通过标准差，方差初步筛除掉变化很小的特征

3.7 PCA降维操作，选取95%的重要性进行保留

pca=PCA(n\_components=0.95)

处理后每个样本特征维度： 439

**4 为数据添加标签**

对target特征‘group‘进行编码，标记为target

**5. 训练模型**  
5.1 逻辑回归模型

clf=GridSearchCV(lr\_model,lr\_param\_grid,cv=5) #网格搜索超参数

5.2 SVM模型

svm\_param\_grid=[{'C':[1e-2,1e-1,1,10,100],'gamma':[0.001,0.0001],'kernel':['rbf']}]

#网格搜索超参数

best\_svm\_model=svm\_grid.best\_estimator\_

best\_svm\_para = svm\_grid.best\_params\_

**6. 预测结果 logloss**

训练逻辑回归模型...

最好的回归模型参数是： {'C': 10}

训练SVM模型...

最好的svm模型参数是： {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}

逻辑回归模型 logloss: 0.5017557850900733

SVM logloss: 0.846481034000951

结果不好，特征工程做的太简单了……