# 易观算法大赛-性别年龄预测

——今晚打老虎 解决方案

线上成绩：2

1. 总体解题思路与算法

我们使用了两套解决方案，产生三个结果，方案A的是基于人工特征基础上应用lgb和DNN神经网络构建两层模型，线上分数是2.57。方案B产生两个结果，RNN 网络和CNN 网络未使用任何人工特征，线上的分数是2.757，然后用lgb模型跑出一些元特征加入NN网络元特征一起跑stacking，线上的分数是2.547。

比赛加权融合方案是：（2.57预测结果\*0.6+2.757预测结果\*0.4）\*0.6+2.547预测结果\*0.4这样产生的线上结果是2.54247，排行榜第二。

方案A的代码是code3文件夹，具体运行步骤见code3文件夹中的readme说明，这个跑出的结果是线上2.57分。方案B的代码是code1文件夹和code2文件夹，要先单独运行code2文件夹代码，这个跑出的结果是线上2.757分。然后运行code2文件夹中的代码，生成元特征和code1的元特征一起根据stacking\_1.py里的配置路径保存，运行stacking\_1.py得到结果取平均得到0.2547的结果。具体运行步骤见code2文件夹中的readme说明。

二，两套解决方案

# 易观算法大赛-性别年龄预测（方案A）

—**今晚打老虎**

目录

[1. 数据预处理 2](#_Toc470195576)

[1.1. 机型数据 2](#_Toc470195577)

[1.2. APP数据 2](#_Toc470195578)

[2. 特征表示 2](#_Toc470195579)

[2.1. Bag of Words 2](#_Toc470195580)

[2.2. TF-IDF 3](#_Toc470195581)

[2.3. SVD 3](#_Toc470195582)

[2.4. 人工构建的特征 4](#_Toc470195584)

[3. 模型结构 4](#_Toc470195585)

[3.1. 第一层模型 4](#_Toc470195586)

[3.2. 第二层融合模型 5](#_Toc470195588)

# 数据预处理

## 机型数据

机型数据中是每个设备品牌和型号信息，机型数据存在的问题主要有一下几点：1.品牌信息混乱，同一品牌的设备没有统一的标识。2.设备的型号信息，部分夹杂着品牌信息。3.设备的品牌和型号特征都有不同程度的缺失。

针对以上的问题，我们进行了简单的数据清洗和缺失值的填充。把属于同一品牌的设备使用统一的字符来表示，缺失的信息用’unkonw’字符串来填充。

## APP数据

APP数据中只包含部分应用的类别信息，同时还有部分的应用类别信息冗余，在后续的特征处理中会有一定的影响，所以对于APP数据，我们进行了两方面的处理，一个是缺失值的填充和冗余类别信息的剔除。

# 特征表示

为了对数据进行更全面的刻画，我们构建了多角度的特征。具体特征如下：

## Bag of Words

通过BOW模型主要是对设备应用安装列表，应用的类别信息以及设备数据中应用的使用情况，通过BOW向量来表示。

## TF-IDF

根据BOW向量计算特征的TF-IDF值。

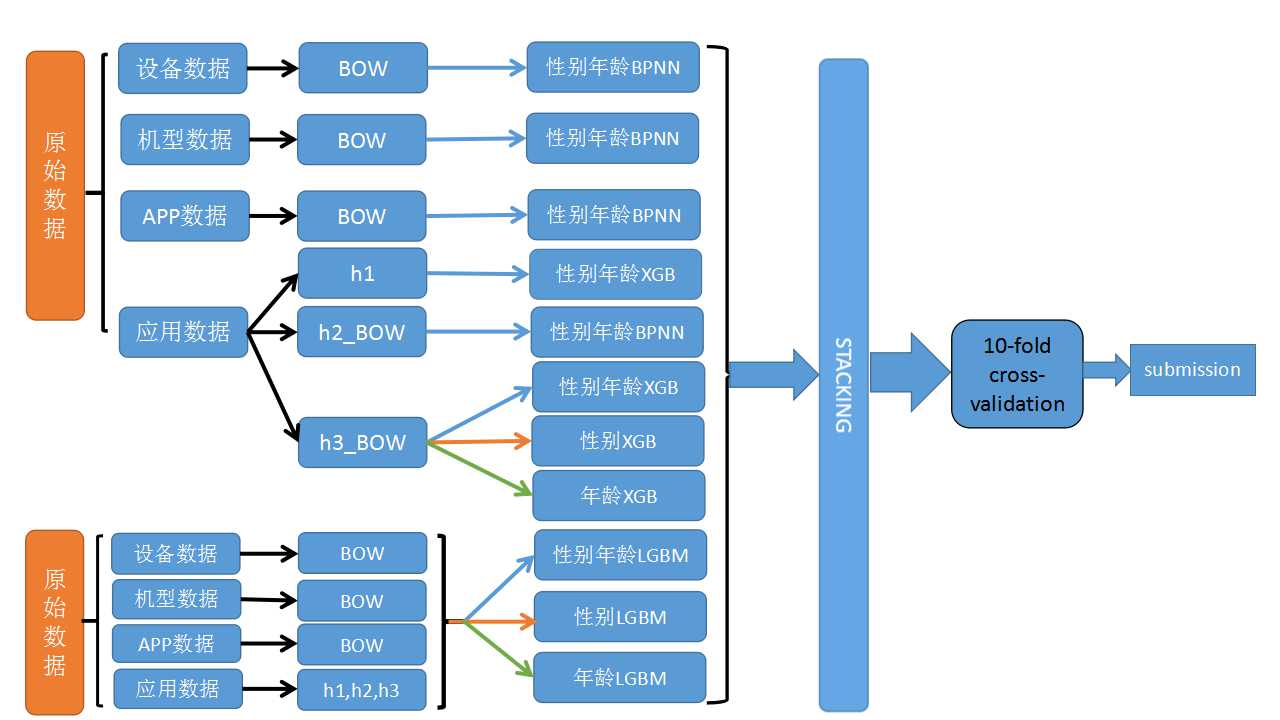
## SVD

通过BOW模型得到的特征维度较高，所以对BOWvec和tf-idf特征进行了降维处理。

## 人工构建的特征

除以上通过模型学习得到的特征表示之外，我们还人工构建了一些特征，人工构建的特征包括：每个设备安装应用数量、使用次数最频繁的应用、使用最频繁应用的类别、应用使用的平均，最大，最小时长、以及每个设备开启应用的最早时段和最晚时段。

# 模型结构



**特征表示** 模型整体结构图

模型采用了两级结构，第一级使用了神经网络模型、XGB、LGBM以及Stacking算法，第二级模型中仅使用了神经网络模型和10折交叉验证的方法，来进一步提升模型的准确性与泛化能力，最后将交叉验证得到的结果进行了融合，生成最终的提交文件。

## 第一层模型

第一层模型中，我们尝试了BPNN、xgboost和LightGBM。其中，由于tfidf和BOW向量特征过于稀疏、维度过高，所以对特征进行了降维处理。在第一层模型中，我们先是对不同的特征数据进行了分组训练，并对个别特征进行了子任务训练。然后将全部特征放在一起进行了训练。同时也使用了Stacking算法来生成用于第二层训练的新数据。

下面表1是各个分类器在第一层模型中的表现，logloss为交叉验证中的平均值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类器 | 特征数据 | 年龄logloss | 性别logloss | 性别年龄logloss |
| BPNN | 设备数据 | \ | \ | 2.61994 |
| BPNN | 机型数据 | \ | \ | 2.86279 |
| BPNN | APP数据 | \ | \ | 2.80120 |
| Xgboost | 应用数据h1 | \ | \ | 2.80071 |
| BPNN | 应用数据h2 | \ | \ | 2.68406 |
| Xgboost | 应用数据h3 | 2.20288 | 0.64545 | 2.84413 |
| LightGBM | 全部数据 | 2.22201 | 0.63944 | 2.64044 |

表1 第一层模型的线下成绩

## 第二层融合模型

第二层模型的输入数据来自第一层模型输出的概率值，在第一层模型中我们分别在年龄性别这个22分类的任务上训练，同时也在年龄和性别这两个子任务上进行了训练。第一层模型输出由8个22分类（年龄性别）任务、2个二分类任务（性别）和2个11分类任务（年龄）输出的概率值组成，所以我们第二层模型输入的特征维数是8\*22+2\*2+2\*11=202。

第二层模型，使用神经网络模型进行了10折交叉训练，然后得到的结果进行融合，得到提交文件。

# 易观算法大赛-性别年龄预测（方案B）

—**今晚打老虎**

解决方案亮点：

1、深度学习网络未使用任何人工特征，深度模型端到端解决问题；

2、计算了每个app使用序列中每个app的 tfidf 特征，对词向量进行优化；

**一、赛题核心难点：**

1、如何利用用户使用app信息数据集提高设备用户信息的预测质量；

2、如何利用品牌信息提高设备用户信息的预测质量；

**二、主要解决思路：**

1、用户使用app信息数据集的利用

1.1 motivation

通过对用户使用app信息数据集进行重新提取，按照用户使用的时间先后顺序进行排序，网络训练之前进行NLP操作对app序列进行词向量训练，将每一个app当成一个word，整个applist当成sentense，得到预训练的app词向量，在网络训练是直接嵌入，对嵌入的embedding 矩阵提取关键app信息，让网络倾向于通过关键词来判断序列对用户信息的分类。

1.2 具体操作

充分利用所有app序列信息数据（包括install和use），首先对全局app序列语料做 tfidf，计算每个app序列中每个app的 tfidf。然后将计算得到的 tfidf 值与app词向量对应相乘， 得到 tfidf 加权后的词向量，最后将新的 embedding 矩阵直接输入到网络中，这样网络中可以不包含embedding层，因为embedding操作在网络外的加权部分已经做完。加权逻辑如 下图1-1所示：

\*

原始embedding矩阵 tfidf向量 修正后的embedding

图1-1

原始 embedding 矩阵 tfidf 向量 修正后的 embedding n\*d 的 embedding 矩阵 通过长度为 n 的 tfidf 向量来修正，修正后的 embedding 矩阵获 取了全局的信息，让网络更加注重关键的单词，有助于对文本相似度的识别。即：

sentence1=[w11,w12,...,w1n] sentence1\_new=[w11\*tfidf11,w12\*tfidf12,...,w1n\*tfidf1n]

1.3、数据增强操作：预测测试集数据，然后挑选置信度极高的样本加入训练集继续训练，效果不明显，耗时较长遂放弃；（这里写的伪标签方案）

**三、网络结构：**

我们的网络结构主要有以下两个：AVRNN 网络和CNN 网络。其余网络均是在两者上的变种。我们尝试过其他更加复杂的基于 attention 的模型，由于app序列过长，会导致模型效率很低，最终没有使用特别复杂的模型。 每个网络会略微调整内部参数及 batch\_size 的大小来产生差异，最后融合增强鲁棒性。 两种网络结构如下图1-2，图1-3所示：

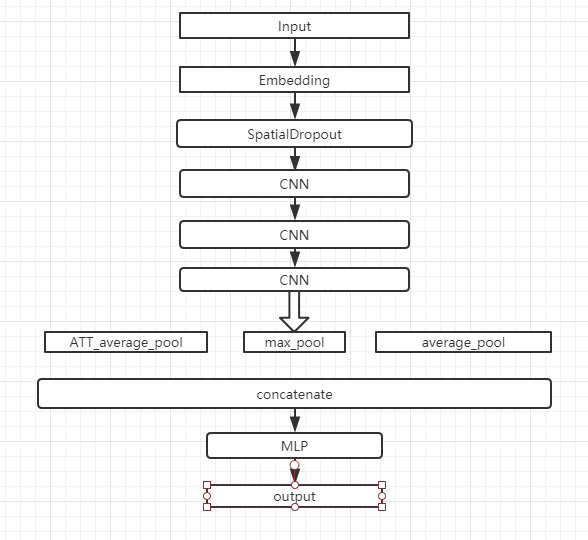


图1-2

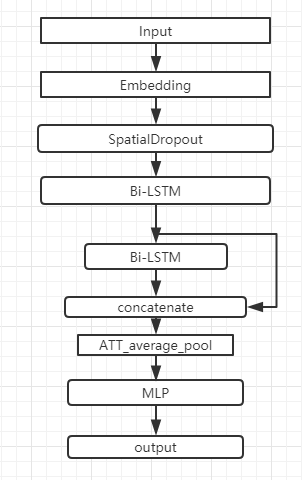


图1-3

**四、其他尝试：**

1、更加复杂的深度模型：数据量较小，过拟合严重；

最后感谢大赛举办方给我们提供这么好的平台和机会。