**华中科技大学**

**计算机科学与技术学院**

**《机器学习》结课报告**



专 业： 计算机科学与技术

班 级： 计卓2001班

学 号： U201914858

姓 名： 王怡彬

成 绩：

指导教师： 何琨

**完成日期： 2022年 6 月 25 日**

目录

[1. 实验题目: 智能家居使用场景识别挑战赛 2](#_Toc107389594)

[2. 实验要求 2](#_Toc107389595)

[2.1 实验任务 2](#_Toc107389596)

[2.2 数据说明 2](#_Toc107389597)

[2.3 评测标准 3](#_Toc107389598)

[3. 算法设计 4](#_Toc107389599)

[3.1 数据处理 4](#_Toc107389600)

[3.2 模型算法 4](#_Toc107389601)

[3.3 模型融合 4](#_Toc107389602)

[4. 实验环境与平台 5](#_Toc107389603)

[5. 实现与分析 5](#_Toc107389604)

[6. 实验结果 8](#_Toc107389605)

[7. 个人体会 8](#_Toc107389606)

# 实验题目: 智能家居使用场景识别挑战赛

品冠科技深耕智能家居多年，基于人工智能和大数据技术，智能家居系统更加智能化，用户的体验也大大提高。

公司智能家居业务发展需要，在全国各地设有不同等级的代理商。为了让用户切身感受到智能家居产品的智能化和便捷性，每个代理商均有自己的智能家居体验店和展厅。

根据公司的发展策略，需要让终端用户在智能家居体验过程中能够更加强烈、清晰的感受到智能家居系统的优点，系统会针对展厅类的场景做特殊的场景优化。为了实现这一场景下的优化，就需要系统能够准确、快速的分析出当前智能家居产品使用环境是真实的家庭还是智能化体验的公共区域。

本次比赛为参赛选手提供了4类数据：账号信息、设备列表、控制操作日志、设备上报日志。其中账号基础数据的训练集会打上使用场景是家庭用户还是体验厅的标签。

出于数据安全保证的考虑，所有数据均为脱敏处理后的数据。数据集会分为两个部分，第一部分数据有使用场景标签，用于识别算法训练，第二部分不包含场景标签，数据用于预测。

# 2. 实验要求

## 2.1 实验任务

根据比赛举办方提供的4类数据：账号信息、设备列表、控制操作日志、设备上报日志预测每个账号的使用场景是家庭用户还是体验厅，即利用表格型数据解决二分类的目标问题，目标为尽可能提高预测结果的F1 Score。

## 2.2 数据说明

比赛为参赛选手提供了4类数据：账号信息、设备列表、控制操作日志、设备上报日志；每个账号都有标明该账号使用场景是家庭用户还是体验厅的标签，而每个账号都会拥有一系列设备，在设备列表（devList.csv）中给出了其对应关系，控制操作日志和设备上报日志说明了每个设备的控制日志和设备日志内容。



图1 数据说明

训练集与验证集的划分方式为训练集大小：验证集大小=7：3，测试集由赛事主办方提供。

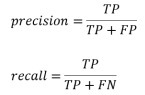
## 2.3 评测标准

由于比赛官方采用了F1 Score作为预测结果的唯一标准进行评价，故在本地验证与模型选择中也采用F1 Score作为主要评测标准。

F1 Score的具体计算结果如下：

（1）统计TP（正确预测环境场景），FP（错将家庭场景预测为公共区域），FN（错将公共区域预测为家庭记录）

（2）通过第一步的统计值计算模型的precision和recall，计算公式如下：



（3）通过第二步计算结果计算F1-score,得到最后评测结果，计算方式如下：



# 3. 算法设计

## 3.1 数据处理

1. 异常值处理

控制操作日志（control.csv）和设备上报日志（devUpdata.csv）中会出现单个设备（单行）拥有额外多列的情况，会导致调用pandas的read\_csv直接失败。根据查看具体异常值发现，多余出的数列均在data列之后，疑似为额外的控制/上传日志信息，介于特征工程中没有处理控制/上传日志信息，故将额外的异常列均参考baseline实现使用了自定义的robust\_readcsv做了删除处理。

1. 特征工程

本次比赛提供的表格型数据为典型的主从表数据，即由一个 user/实体 的基本信息表（账号信息）、以该user/实体为主键的多个历史行为表（控制操作日志、设备上报日志）和由主键与历史行为表之间的串联表（设备列表）组成。而因此特征工程的主体内容即为对历史行为表中的时序特征进行特征工程encoding。

（1）对于设备列表，我们对于其‘type’及‘area’两列文本特征使用tfidf转化为了数千个特征列表示该user在各分词上的tf-idf指数，再将测试集中未出现的分词对应的特征列删除，最终只剩下了三百多列特征。同时也对每个user统计了‘type’和‘area’的去重后元素的个数（以下简称nunique）、 非空数值个数（以下简称count），‘did’的nunique属性加入特征。

（2）对于历史行为表（控制操作日志、设备上报日志），我们对于每个设备统计了'did'、'form'（对于设备上报日志、无该列），'data'各自的nunique属性，每周的平均count和每日的平均count，最早行为时间与最晚行为时间的差值，每条行为与上条行为的平均时间差值。再对每个user，将其所有控制设备的这些统计量加和，加入特征。

3. 缺失值处理

由于特征工程中对特征量的统计主要为频率和计数统计，故将缺失值均以0填充。

4. 特征清洗

直接使用adaboost学习出的特征重要性（feature\_importance），以加权平均（quantile）为依据筛选出了quantile大于0.2的特征作为之后adaboost与tabnet的输入特征。

## 3.2 模型算法

**3.2.1 TF-IDF**

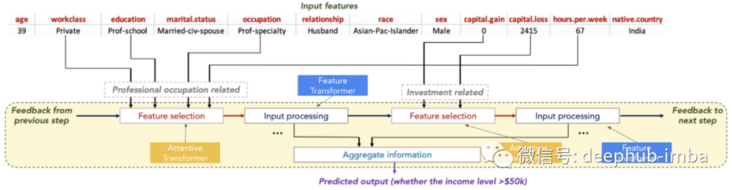
随便抄点

**3.2.1 TabNet**

Google发布的TabNet是一种针对于表格数据的神经网络，它通过类似于加性模型的顺序注意力机制（sequential attention mechanism）实现了instance-wise的特征选择，还通过encoder-decoder框架实现了自监督学习。

TabNet 使用 Sequential Attention 的思想模仿决策树的行为。简单地说，可以将其视为一个多步神经网络，在每一步应用两个关键操作：Attentive Transformer 选择最重要的特征在下一步处理；通过Feature Transformer 将特征处理成更有用的表示

模型最后使用Feature Transformer 的输出稍后用于预测。TabNet 同时使用 Attentive 和 Feature Transformers，能够模拟基于树的模型的决策过程。例如以下的成人人口普查收入数据集的预测，模型能够选择和处理对手头任务最有用的特征，从而提高可解释性和学习能力。

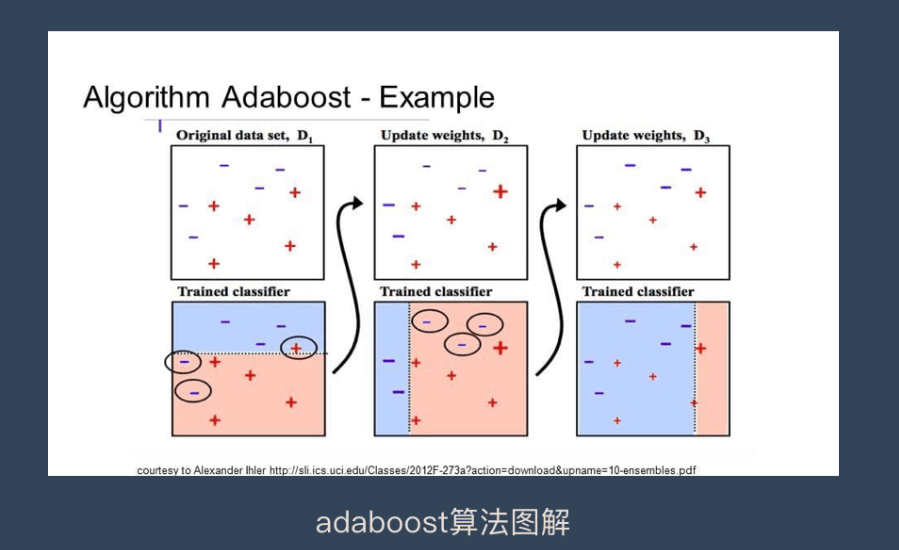


Attentive 和 Feature Transformer 的关键构建块是所谓的 Feature Blocks

**3.2.2 Adaboost**

Boosting是一种集合技术，试图从许多弱分类器中创建一个强分类器。这是通过从训练数据构建模型，然后创建第二个模型来尝试从第一个模型中纠正错误来完成的。添加模型直到完美预测训练集或添加最大数量的模型。

AdaBoost是第一个为二进制分类开发的真正成功的增强算法。这是理解助力的最佳起点。现代助推方法建立在AdaBoost上，最着名的是随机梯度增强机。



AdaBoost用于短决策树。在创建第一个树之后，每个训练实例上的树的性能用于加权创建的下一个树应该关注每个训练实例的注意力。难以预测的训练数据被赋予更多权重，而易于预测的实例被赋予更少的权重。模型一个接一个地顺序创建，每个模型更新训练实例上的权重，这些权重影响序列中下一个树所执行的学习。构建完所有树之后，将对新数据进行预测，并根据训练数据的准确性对每棵树的性能进行加权。

因为通过算法如此关注纠正错误，所以必须删除带有异常值的干净数据。

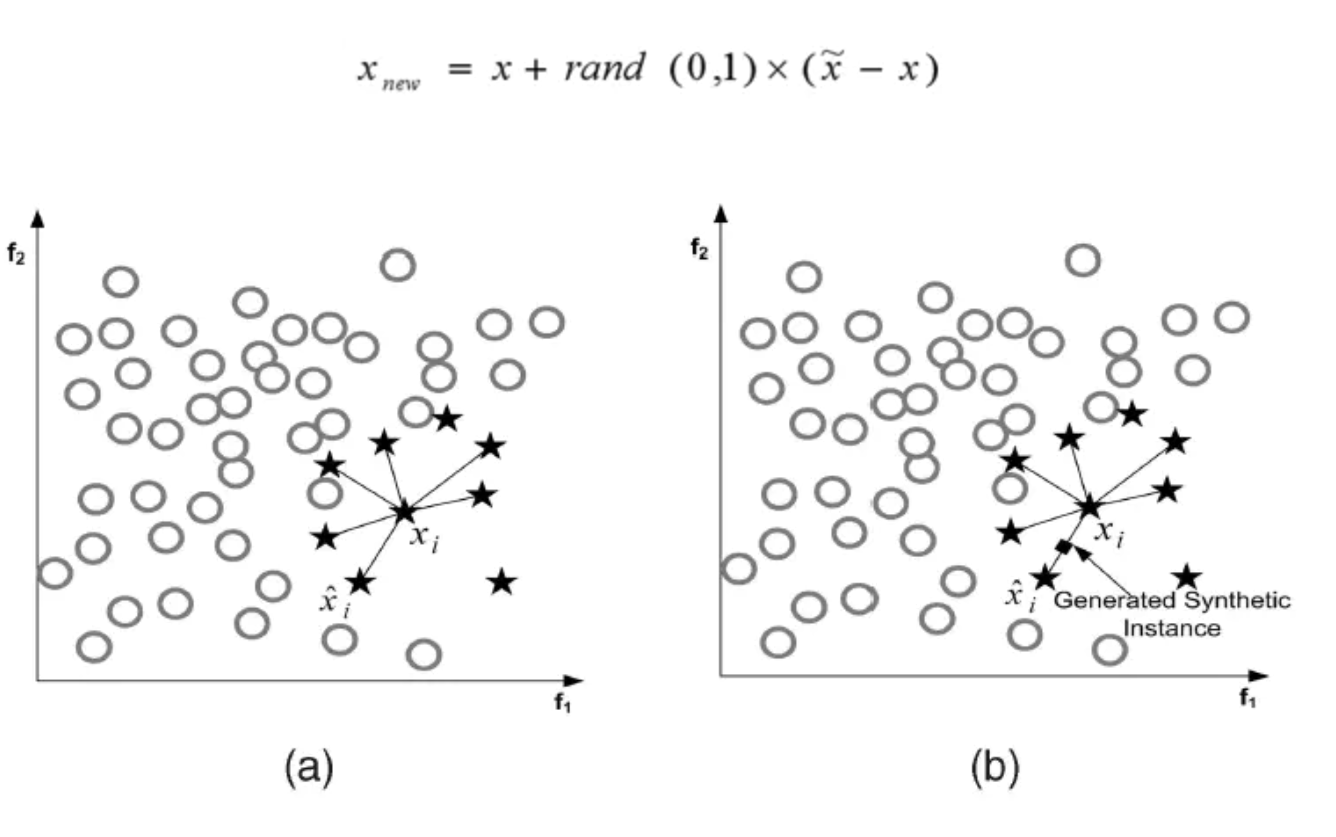
**3.2.3 SMOTE**

SMOTE（Synthetic Minority Oversampling Technique），合成少数类过采样技术．它是基于随机过采样算法的一种改进方案，由于随机过采样采取简单复制样本的策略来增加少数类样本，这样容易产生模型过拟合的问题，即使得模型学习到的信息过于特别(Specific)而不够泛化(General)，SMOTE算法的基本思想是对少数类样本进行分析并根据少数类样本人工合成新样本添加到数据集中，具体如下图所示，算法流程如下。

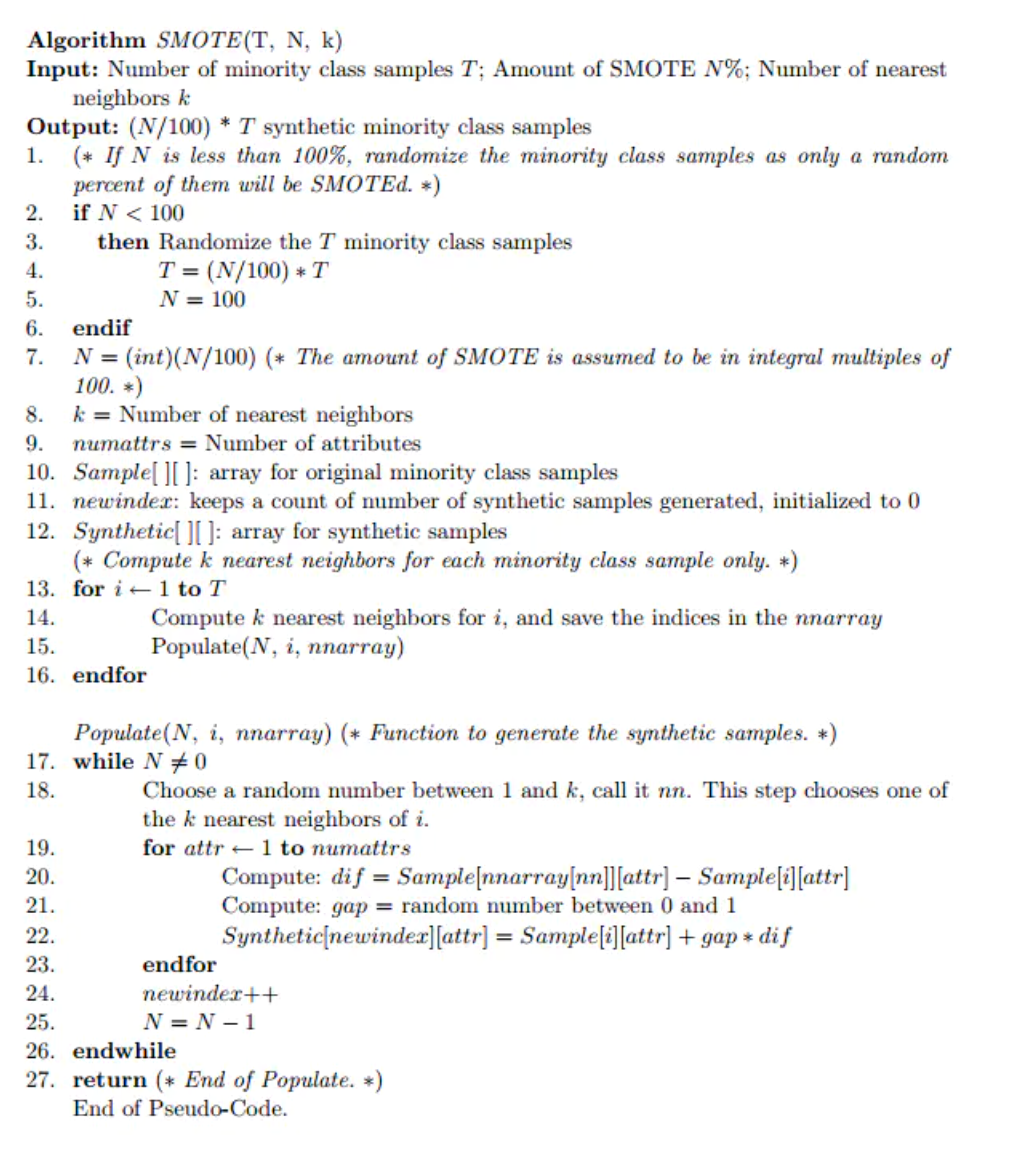
(1)对于少数类中每一个样本x，以欧氏距离为标准计算它到少数类样本集中所有样本的距离，得到其k近邻。

(2)根据样本不平衡比例设置一个采样比例以确定采样倍率N，对于每一个少数类样本x，从其k近邻中随机选择若干个样本，假设选择的近邻为xn。

(3)对于每一个随机选出的近邻xn，分别与原样本按照如下的公式构建新的样本。



smote算法的伪代码如下：

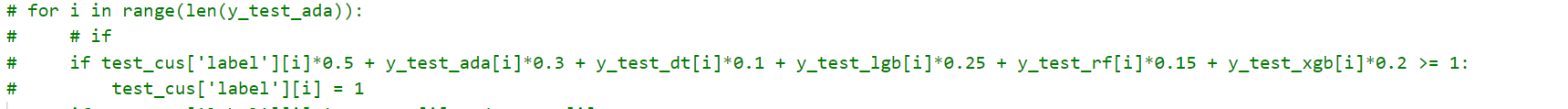


最终模型：Tabnet+”GAN”+数据预处理部分的SMOTE算法

尝试过的模型以及算法：Adaboost, LGBM, autodl自动调参，决策树，Random Forest，甚至贝叶斯、逻辑回归等算法。

最后选择了神经网络tabnet+“GAN”。

## 3.3 模型融合



前期尝试过模型融合，效果并没有那么好，不像之前打的比赛，涨点不明显，同时我们依赖的precision还有所下降，虽然我们的

# 4. 实验环境与平台

Kaggle , Python=3.7,

Python版本，GPU, CPU, 自己的PC或者云平台

随便写写

# 5. 实现与分析

这里的格式同学们自由发挥了，举个例子：关键模块的实现，以及这个实现对于得分的长进，不要贴大段代码（每段代码尽量别超过半页）

鼓励同学们对不同设计的结果进行分析，思考为什么A不如B

这里也可以做一些好看的可视化，比如超参选择和得分的关系什么的

对于多人组队的同学，多写一个成员分工

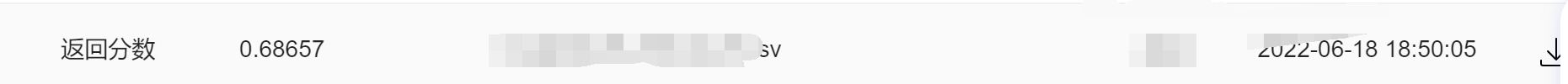
成员分工：

共两人逯润雨，王怡彬，无明显分工，均参与代码编写思考讨论，贡献相同。

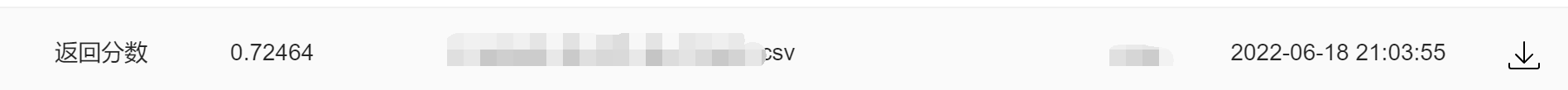
说明，这里我们两个人分别注册了一个账号，所以id是两个图的id

模型主要使用了Adaboost，同时也尝试使用了非常多种的模型融合，比方说LGBM，但是发现这个模型效果并不是很好，曾经也尝试使用一些其他的像逻辑回归、贝叶斯的算法boosting, stacking, bagging，但是效果提升并不明显。

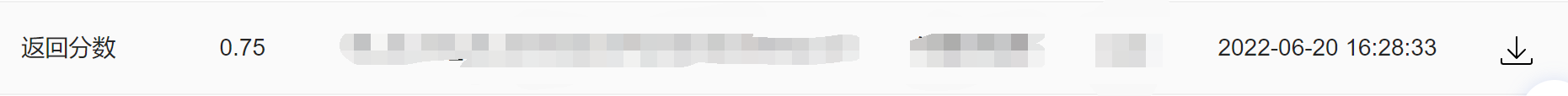
不过起码有一个基本的base效果。没有调参跑基本的baseline提交上去的结果如下：



有一个基本的0.72的baseline还不错，后来我们选择使用了好多种模型算法同时进行bagging,stacking,boosting各种调参大法（不是）疯狂调参提交了几天，发现效果很多时候还不如之前的效果，后来使用adaboost然后做了一些科学的调参：交叉验证折数提升，autodl搜索，设定层深度为4，调了几次得到了一个还算比较满意的效果：



后来调整的时候注意到了或许我们需要做一些样本的均衡操作，因为实际上训练样本中实际上正例是非常少的，当然测试样本经过测试正样本也不多，但是测试样本的正样本比例还是要高很多的，所以我们，我们大致提交了一次全一的效果大致算了下正负类占比然后通过调整训练做了。。。。。。。。（待wyb补充）的调整我们得到了一个最好差不多能达到0.75的结果：



由于比赛直接使用F1 Score作为唯一评测标准，我们可以直接使用特殊提交结果测出测试集正确预测结果中正类占比，将提交的预测结果全部置为1（即全预测为正类），根据提交结果得知为F1=0.2549。根据F1 Score的计算公式得到，若提交记测试集正确预测结果中正类占比为 p，则F1=2p/1+p=0.2549，可直接解得测试集正确预测结果中正类占比p≈0.146，而测试集样本数n=267，故算得测试集正确预测结果中正负类比例为正类；负类=39；228。

此时再回看提交结果，发现adaboost的precision达到38/39好像是，只记得错了一个。由此确定投票的一个基本策略之一：if y\_ada[i] = 1: y\_test[i] = 1。这种情况直接一票通过。

这里说明下为什么不使用提分神器lgbm，主要是数据集就这么大并且训练速度都差不多并且 很明显这里在我们测试的任何时候lgbm的效果分数都不如adaboost，甚至不如决策树有时，所以就选择不使用lgbm。

由于我已经知道了39（）个正确正类，故可以一方面调整adaboost的n\_estimaters,另一方面用tabnet进行针对性的调优，经过后面的提交结果可得知tabnet能保持一定精度的同时保持在0.75、0.76左右的recall（recall是通过F1反过来算的）。

于是：

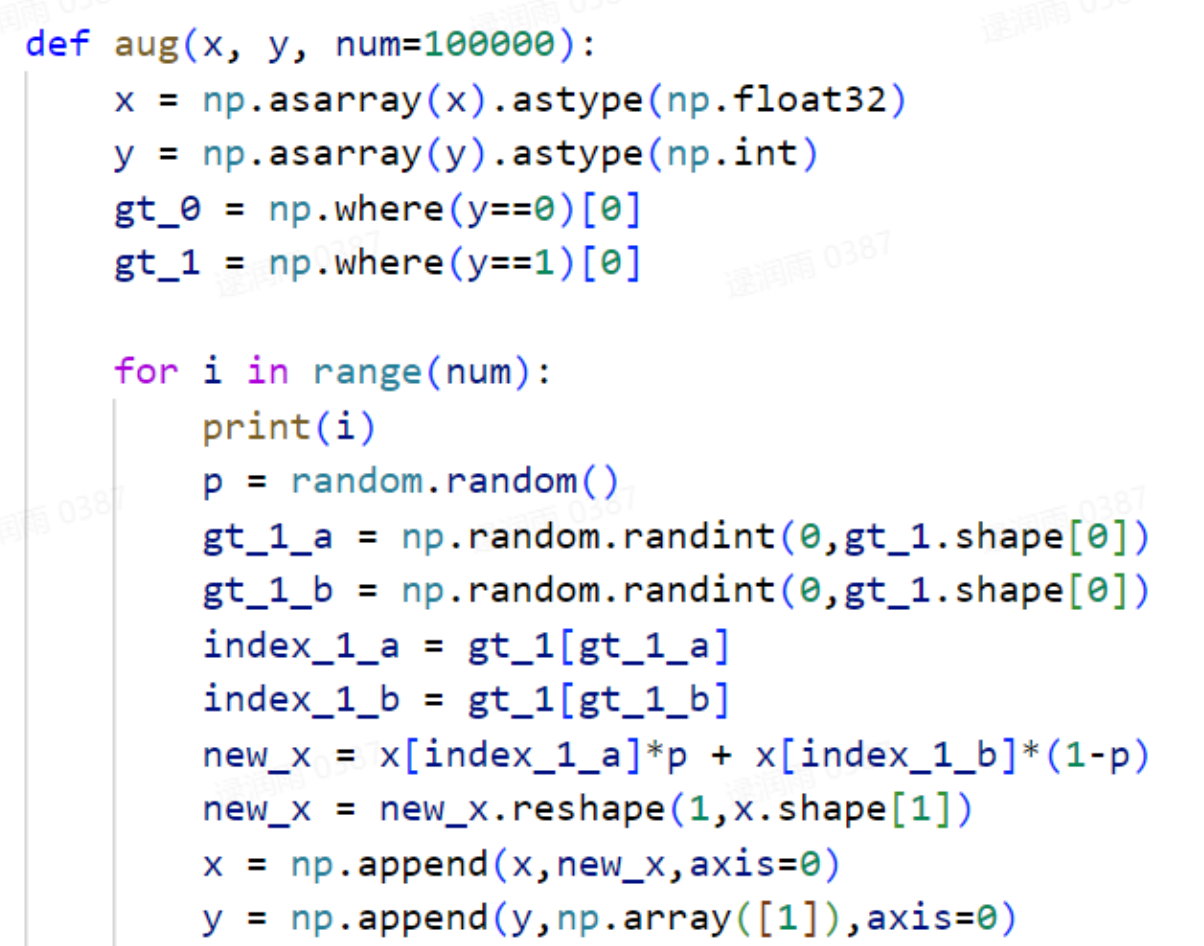
from pytorch\_tabnet.tab\_model import TabNetClassifier（tabnet接口，好用，但是输入需要pd.Dataframe2torch.Dataset）

y\_test = tabnetmodel.predict(x\_test)

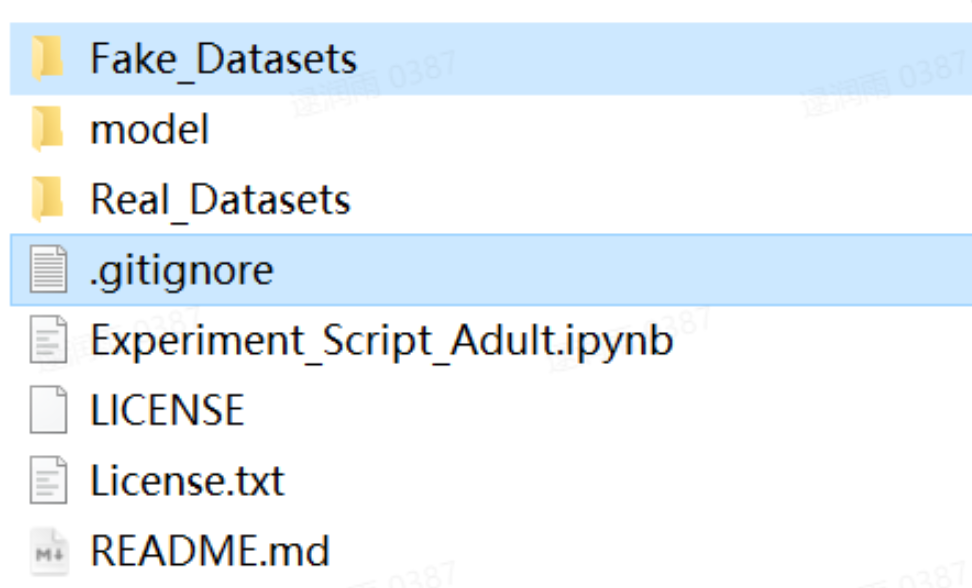
if y\_ada[i] == 1:

y\_test[i] = 1

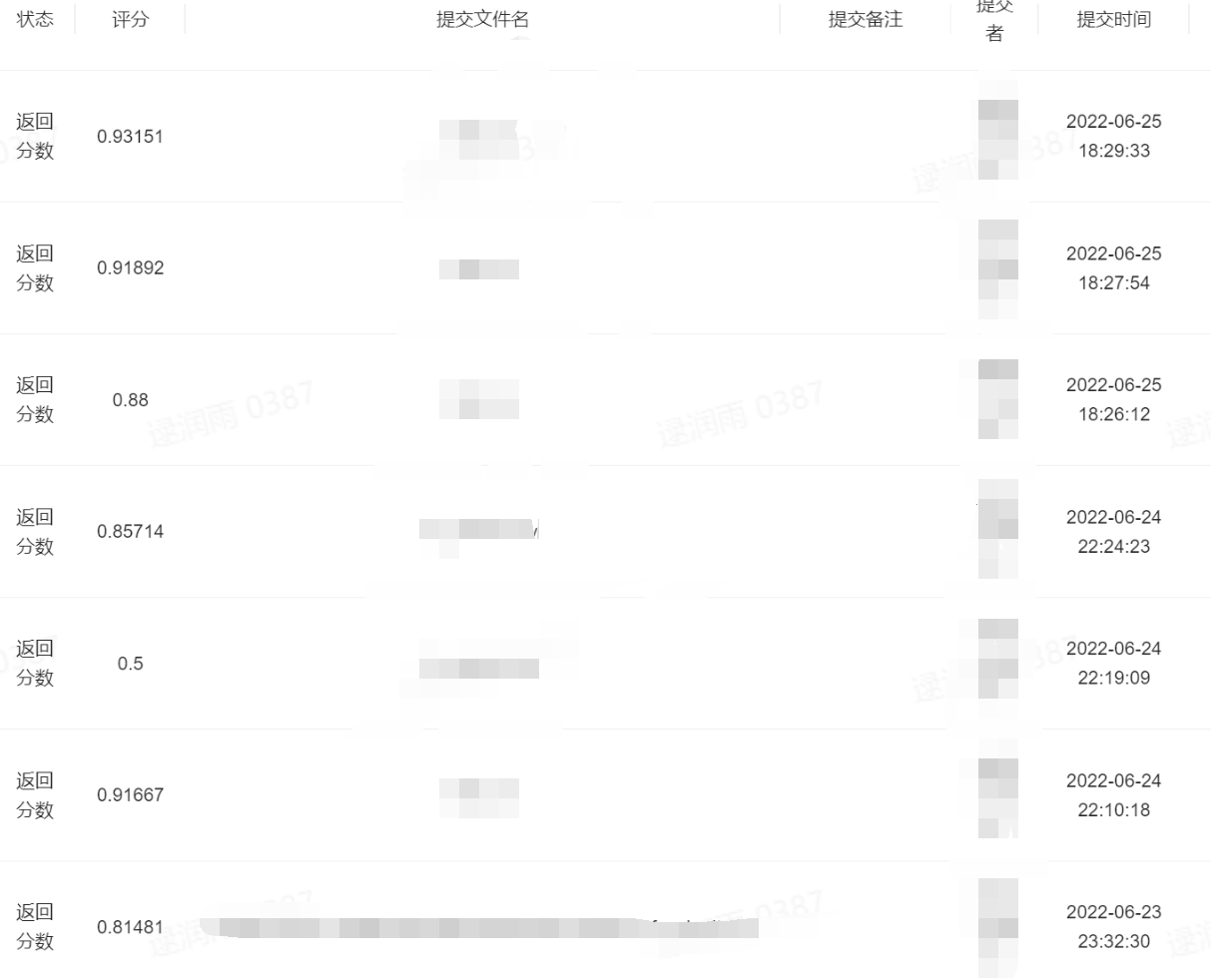
定义一个类似下图的函数来产生第一批fake\_dataset



由于可以通过每次提交都能算出TP，故能确定的项会递增。故第二版fake\_dataset可由y\_test里的确定项通过上面aug函数得出，这一步至关重要



接下来就不断扩充两个数据集，不断迭代提升精度一步步往上提升精度就可以了！基本上后期因为确定的很多了很多分数我们都能直接算出来，所以我们就没有提交，到后来确定会有提升才会将结果提交上去。



最后很幸运的拿了榜一。

# 6. 实验结果

截止2022.6.28， 智能家居使用场景识别挑战赛初赛第一名， 得分为0.93151，比第二名高0.7分，第三名高0.8分。



# 7. 个人体会

对课程建议：

建议之后不要设定打比赛这类作业，因为临近考试周这种几乎没有上限、不确定性太高的比赛没有很让人信服的评分标准以及需要花费大量时间调参并且不知道会不会有效果，这导致没时间复习别的考试内容并且这门课分数也不高，对于很多人不公平并且对于机器学习本身算法学习没啥用。建议可以手写一个决策树这类作业，另外建议之后作业难度低一点，手写这些框架非常非常花时间，可以是调用一些框架做些有意思的东西。比方说预测什么东西之类的。