**赛题说明**

信用风险评分 [收起题目详情](javascript:void(0))

一、赛题背景

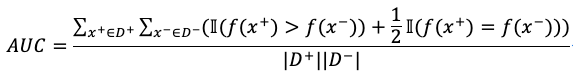
在当今大数据时代，信用评分不仅仅用在办理信用卡、贷款等金融场景，类似的评分产品已经触及到我们生活的方方面面，比如借充电宝免押金、打车先用后付等，甚至在招聘、婚恋场景都有一席之地。

招行作为金融科技的先行者，APP月活用户数上亿，APP服务不仅涵盖资金交易、理财、信贷等金融场景，也延伸到饭票、影票、出行、资讯等非金融场景，可以构建用户的信用评分，基于信用评分为用户提供更优质便捷的服务。

二、课题研究要求

本次大赛为参赛选手提供了两个数据集（训练数据集和评分数据集），包含用户标签数据、过去60天的交易行为数据、过去30天的APP行为数据。希望参赛选手基于训练数据集，通过有效的特征提取，构建信用违约预测模型，并将模型应用在评分数据集上，输出评分数据集中每个用户的违约概率。

三、评价指标



其中D^+与D^-分别为评分数据集中发生违约与未发生违约用户集合，|D^+ |与|D^- |为集合中的用户量，f(x)为参赛者对于评分数据集中用户发生违约的概率估计值，I为逻辑函数。

四、数据说明

1.训练数据集\_tag.csv，评分数据集\_tag.csv提供了训练数据集和评分数据集的用户标签数据；

2.训练数据集\_trd.csv，评分数据集\_trd.csv提供了训练数据集和评分数据集的用户60天交易行为数据；

3.训练数据集\_beh.csv，评分数据集\_ beh.csv提供了训练数据集和评分数据集的用户30天APP行为数据；

4.数据说明.xlsx为数据集字段说明和数据示例；

5.提交样例：

5.1采⽤UTF-8⽆BOM编码的txt⽂件提交，⼀共提交⼀份txt⽂件。

5.2输出评分数据集中每个用户违约的预测概率，输出字段为：用户标识和违约预测概率，用\t分割，每个用户的预测结果为一行，所有数据按用户id从小到大排序，注意不能有遗漏的数据或多出的数据。

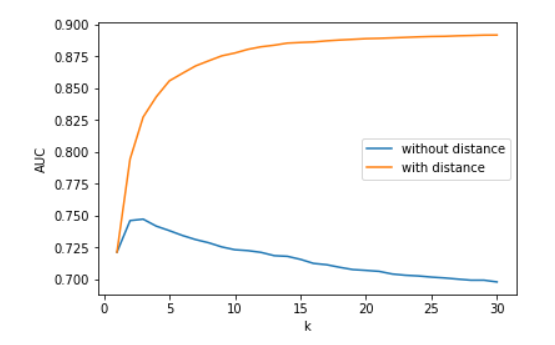
上传预测结果数据，格式仅限txt文件，大小不得超过10M

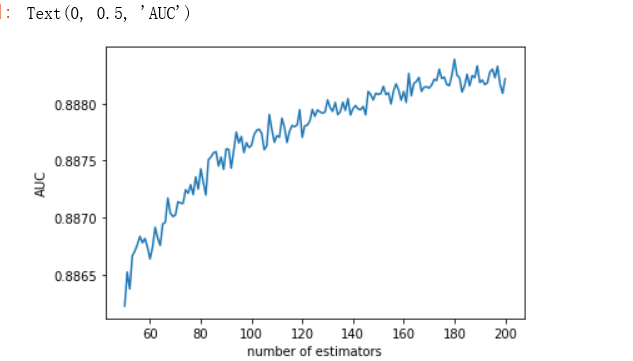
点击上传代码附件，仅限zip格式。（不超过100M）

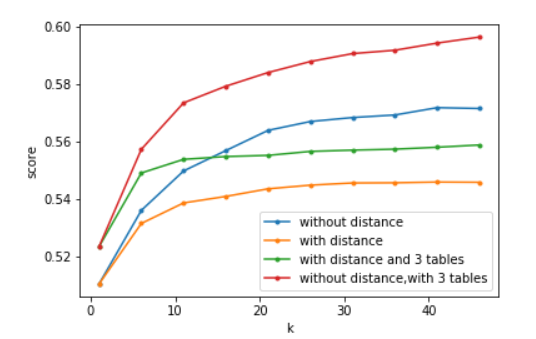
针对用户行为 交易可以采取统计特征的方法，即mean, median, std等

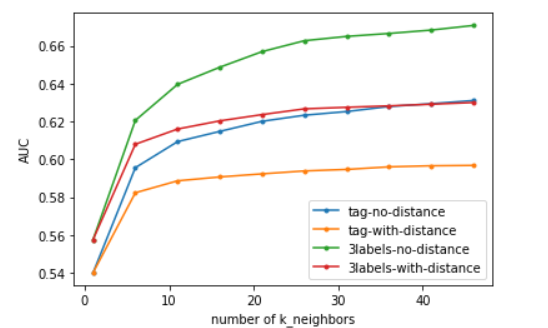
常见BUG : df.loc [df[label]>n] = 3!!!

df.loc [df [label] > n, label ] = 3

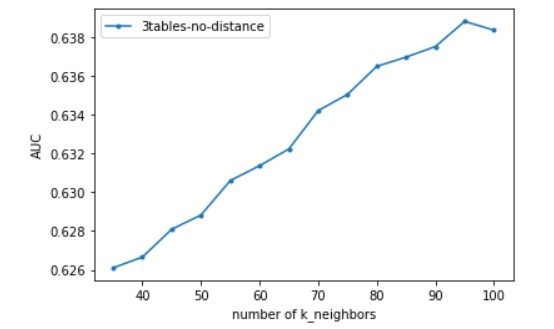








后续改进：重新考虑对缺失值有较大影响的feature以此来重新填充null



目前为止 k = 95 最佳

2020/5/4

对非缺失feature作分组处理

在分组处理前，用KNN对缺失值进行填充，且含有缺失值的列不作分组处理

仅仅把 attd\_type, deg\_cd, edu\_deg\_cd drop

使用page\_no和 trx cny\_trx\_amt，分别用KNN和自定义方法填充

最终用 random\_forest（n\_estimators = 100）得到最佳成绩 0.67,本地0.609

本地n = 150 auc更高，但上传后测试却只有0.657

2. 对购买理财产品次数进行one-hot-encode，即转为1，0表示过去12个月是否购买过

同时把trx\_cny\_amt, trx\_std 进行标准化，随后cross\_val\_score 0.61提高了0.001

xgboost 本地 0.627 比获得最高的randomforest要高，上传后是0.65842

3. KNN fill null时，把 flag 也加入 x\_label里面 （没有显著提升,0.65655）

4. 下一步还是得进一步进行 feature engineering, 由之前的经验看以看出，drop看似无关feature对AUC影响很大，说明这些 feature并不是无关的，相反之前没有drop的feature可能其实并不相关，可以考虑drop掉，另外trade表对AUC影响也较大，可以进一步加工，而且有些用户没有绑定app，所以不会有app行为，应该都填充为1，可以把几个关系到违约到page\_no提取出来转为1，0

5. 通过对page\_no 进行假设检验，选取出了4个page，rfc score=0.66

然后还尝试了drop掉 'hav\_hou\_grp\_ind','frs\_agn\_dt\_cnt','his\_lng\_ovd\_day' 几个feature

然而 rfc socre=0.659，还略有下降

xgboost local csv=0.63 ，上传后0.66, drop feature后也是下降了

6. drop掉 

后 本地cvs都提高了，xgboost=0.648,本地最高，然而上传后也只有0.648

rfc上传后下降至0.657,说明cards还是相关的

6. 对 app行为表中所有的页面以及性别、学历、婚姻采取one hot encoding，这样做对于LR的计算更加有利 LR cross\_val\_score=0.638

同时这样做导致数据维数增大了很多可以考虑PCA降维处理