参赛队名：404 大队
A 榜成绩：0.75461 （第 3）
B 榜成绩：0.75493 （第 9）
复赛： （第 4）
决赛： （第 4）

竞赛方案报告书

一、赛题分析与理解

1.赛题背景与任务

在”好人贷“的量化风控实践中，四川新网银行面临多个维度的挑战：**高维数据、稀疏数据、无标签样本、多产品客群好坏样本不平衡**等等。比赛提供真实业务场景下的脱敏数据，包含多产品（客群）的高维特征数据和表现数据（部分有标签，部分无标签），邀请参赛者对数据进行探索分析，综合利用监督和半监督机器学习算法、迁移学习算法等设计**区分能力高、稳定性强**的信用风险预测模型，对客户信用风险进行预测。

2.赛题数据与解析

此次竞赛提供的数据包括用户 id，157 项**脱敏的属性/行为特征**，以及是否属高风险用户的标签项。一共有 3 个文件：

1.train_xy.csv，带标签的训练集数据，共 15000 条

2.train_x.csv，不带标签的训练集数据，除无标签字段‘y’外，其余字段与 train_xy.csv 相同，共 10000 条

3.test_all.csv，测试集数据，除无标签字段‘y’外，其余字段与 train_xy.csv 相同，共 10000 条

注：

①特征变量名称以“x_”开头， 其中，特征变量 x1-x95 是数值型变量，x96-x157 是类别型变量，x 变量的缺失值统一以-99 表示。

②标签变量名称为“y”，**0 代表低风险客户，1 代表高风险客户**。

③数据集中共包括 3 个代表不同产品或不同特征的客群信息，客群标示变量名为“cust_group”，不同取值代表不同客群客户。请注意，在本次比赛中，请不要将该变量作为模型训练特征变量。

根据赛题任务与数据，可以将问题转化为“**二分类**”问题，0 代表低风险客户，1 代表高风险客户，0:1 比例约为 21:1（且不同 cust_group 的样本分布也不平衡），评估指标为 AUC，并且提供有/无标签数据样本，可以使用**监督与半监督**方法综合预测用户的信用风险概率。

二、数据清洗与处理

1.缺失值分析

根据数据介绍，x 变量的缺失值统一以-99 表示，我们首先对每个 X 特征变量，在**列方向**上，进行缺失值个数统计，从而转化为**缺失率**，表明某个特征缺失是否严重。下图以 train set 为例，作图如图 1。可以看出，有较多的特征缺失率高达 99%，说明这些特征缺失严重，可能会对模型预测带来干扰。

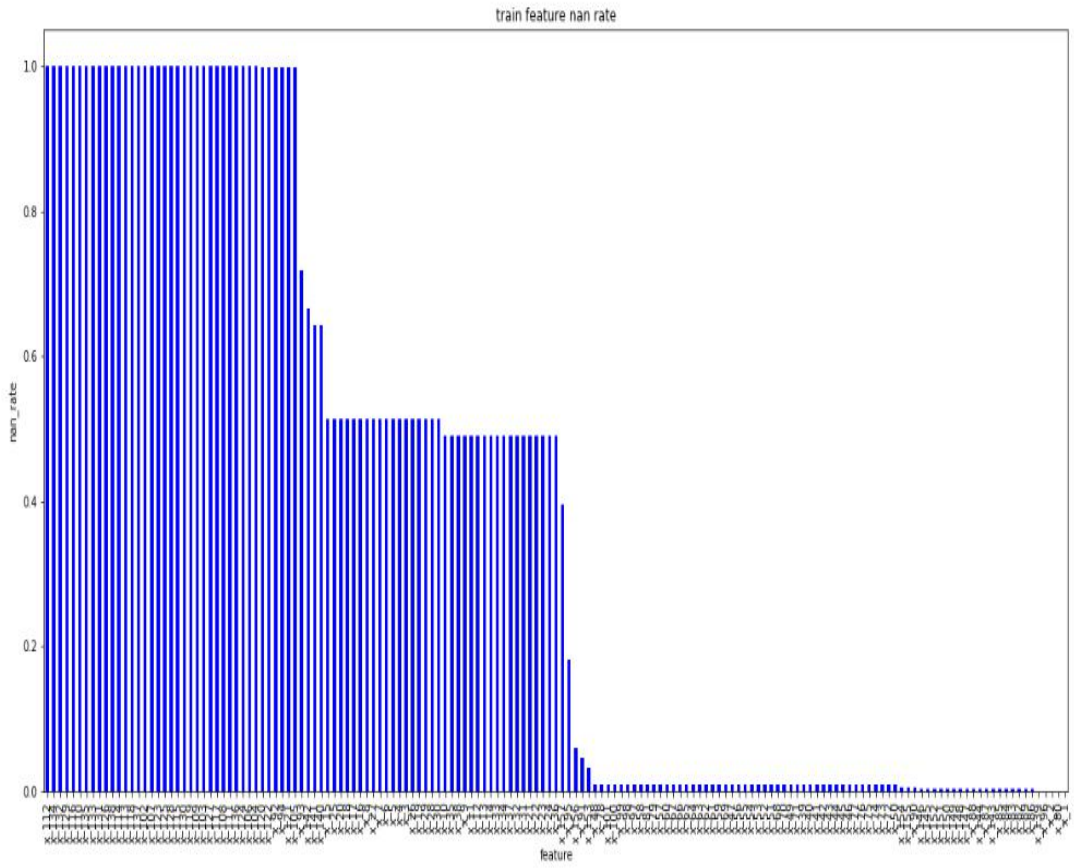


图 1：train set 的特征缺失率统计图

一般来说，对于高维数据和普通模型，通过删除缺失率较高的特征，可以减少噪音特征对模型的干扰，但是我通过 xgb(XGBoost)和 lgb(LightGBM)等树模型训练数据发现，直接删除缺失严重的特征会稍微降低预测效果。这是因为树模型自己在分裂节点的时候，会**自动选择特征**，确定特征的重要性，那些缺失严重的特征，重要性会等于 0。这就像 L2 正则化一样，对于一些特征进行惩罚，使其特征权重等于 0。于是我用全部数据和特征，训练 train set，观察特征重要性作图如图 2，通过对比发现，缺失值严重的特征基本重要性都为 0，但是也有一两个特征重要性不为 0。所以实验表明，直接删除缺失严重的特征，会误删一些对模型有些许效果的特征，而不删除，其实对于树模型来说，影响不大。

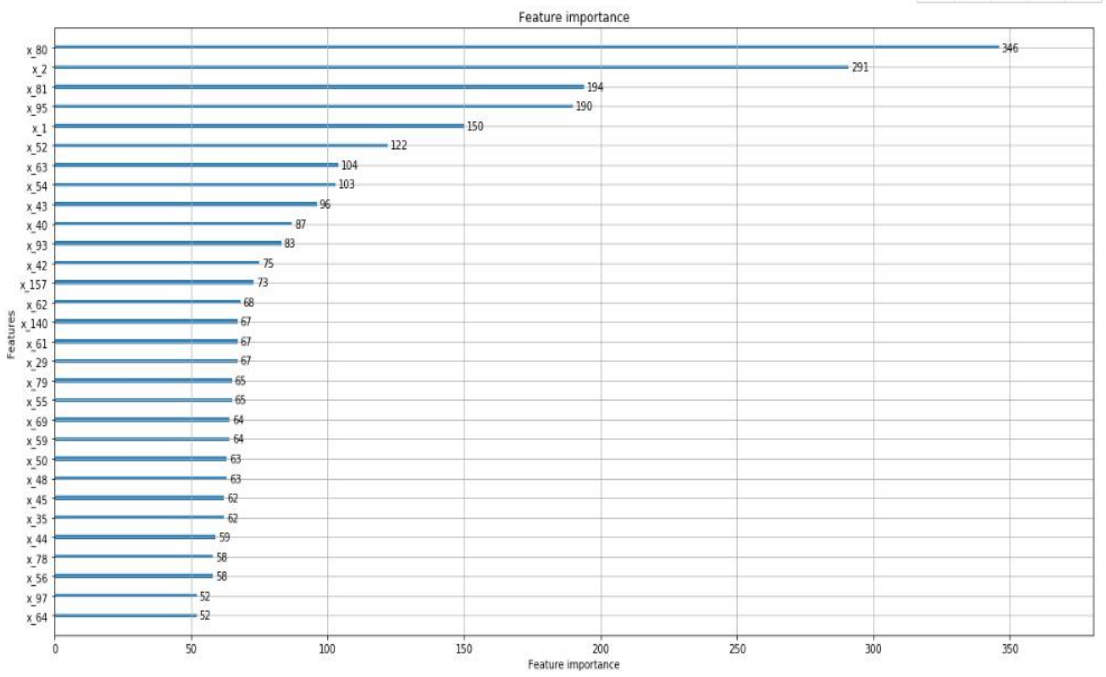


图 2：lgb 模型训练 train set 的 top30 特征重要性

所以，我最后决定不删除任何原始特征，而是使用模型自动选择。这样做，有两个考虑，一是：特征维度并不高（157 维），而且是**匿名特征**，很难确定具体含义；二是：模型自身具有选择特征的特性，可以更好的表现数据。

2.缺失值处理

根据上面的分析，几乎所有数据都存在缺失值-99。一般处理缺失值的方法有**中位数、平均数、众数填充**等操作。但是数据缺失严重的情况下，盲目填充等于增加数据噪声，一股脑填充所有特征是不可取的。所以，我们根据图 2 模型特征重要性的反馈，来观察这些特征数据为什么表现的重要？

①x_1, x_2, x_80 三个特征**没有缺失值，表现最好**

```
# 现在看看一些重要的特征
# top 1-5 :'x_80', 'x_2', 'x_81', 'x_95', 'x_1',
train['x_1'].plot(kind = 'kde') # float
```

```
<matplotlib.axes._subplots.AxesS
```

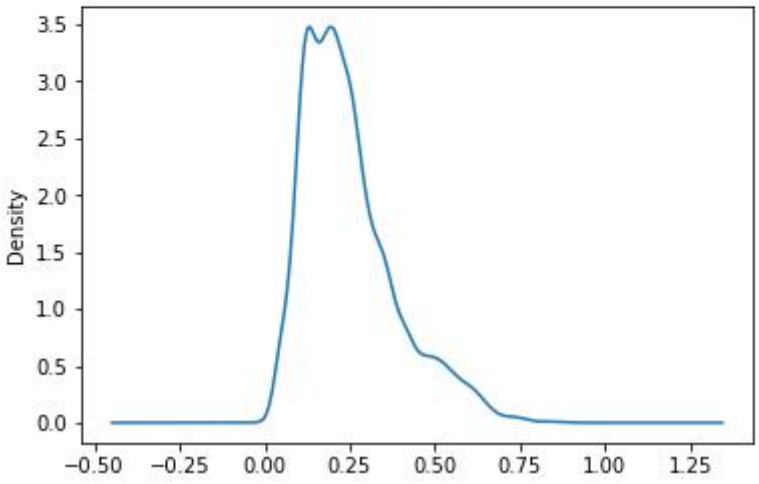


图 3：x_1 的密度图

②x_81, x_95 **缺失较少**，而且**非缺失数据分布密集**

```
train['x_81'].plot(kind = 'kde')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba5bd8ee10>
```

```
train['x_95'].plot(kind = 'kde')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba5bd0bd30>
```

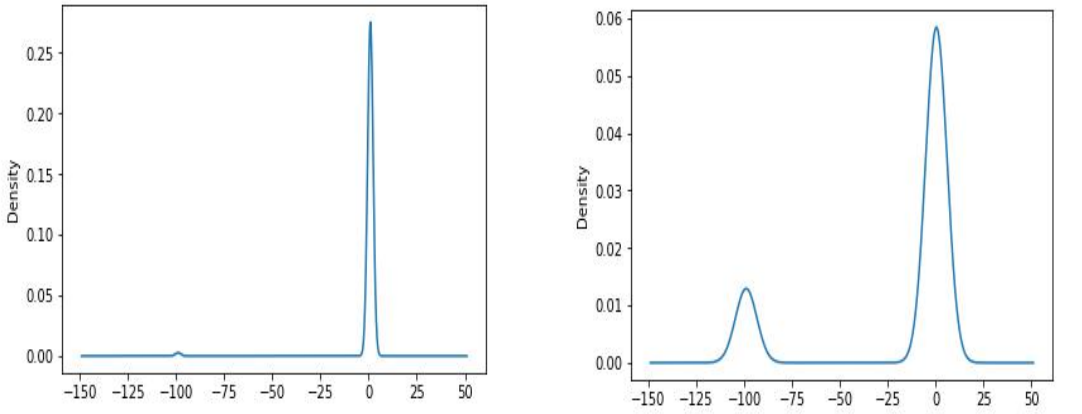


图 4：x_81, x_95 密度图

所以尝试对 x_81,x_95 进行**均值填充**（注：**非缺失值的数据的均值**），效果如下：

```
df1 = train['x_81'].replace(-99,np.nan)
df1.fillna(df1.mean()).plot(kind = 'kde')
```

```
<matplotlib.axes._subplots.Axes
```

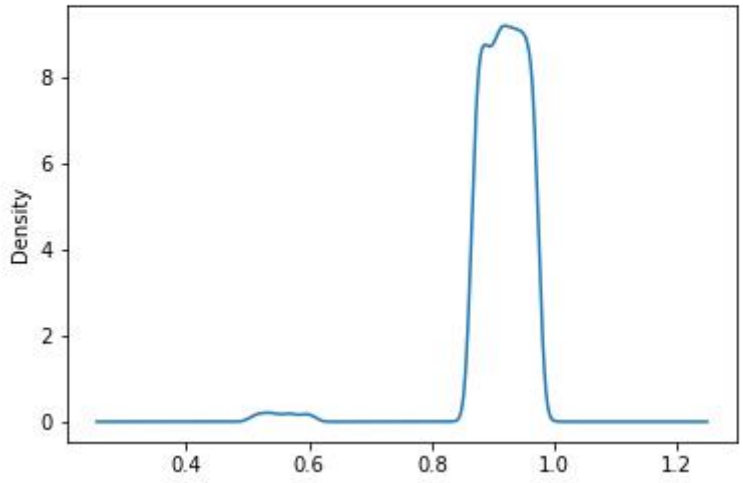


图 5：x_80 均值填充

根据效果图，可以看出，对重要的数值特征进行均值填充，具有不错的效果，实际操作中，对**重要性靠前的 topK** 的数值特征进行均值填充线上线下提升 0.002（K 取 5,10,15,20,30,50，最后实验表明，K = 10 效果最好）。

接下来分析缺失的类别特征，首先观察重要性靠前的类别特征

③类别特征 x_157,x_140，**缺失值-99 的情况下，y=1 最多**。

```
pd.crosstab(train['x_157'],train['y']).plot(kind = 'bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f94ff719898>
```

```
pd.crosstab(train['x_140'],train['y']).plot(kind = 'bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f950c176470>
```

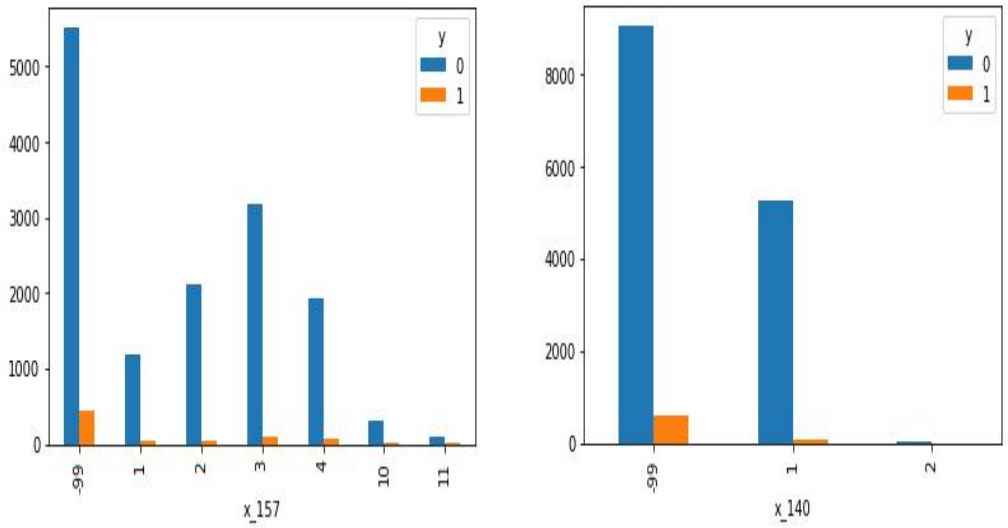


图 6:x_157,x_140 类别特征柱状图

通过作图会发现，重要的类别特征下，当 x_i = -99 时，y=1 很多，这说明，该**特征为缺失值-99 的表现能力很强**，为了验证这一点，我将每个 X 特征是否为-99（是-99 则为 1，否则为 0）作为一列特征，训练全部 train set 数据，得到重要性排序如下：

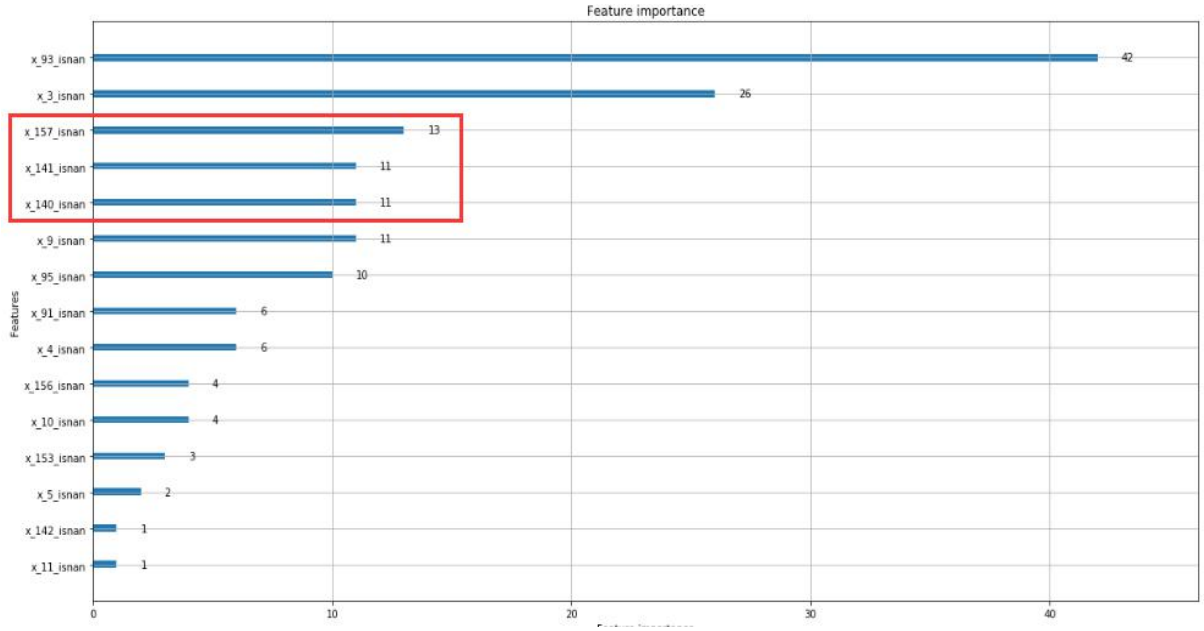


图 7：train set 中特征是否为-99 的重要性排序

并且对所有类别特征进行 onehot 编码，训练 train set 所有数据得到特征重要性如下：

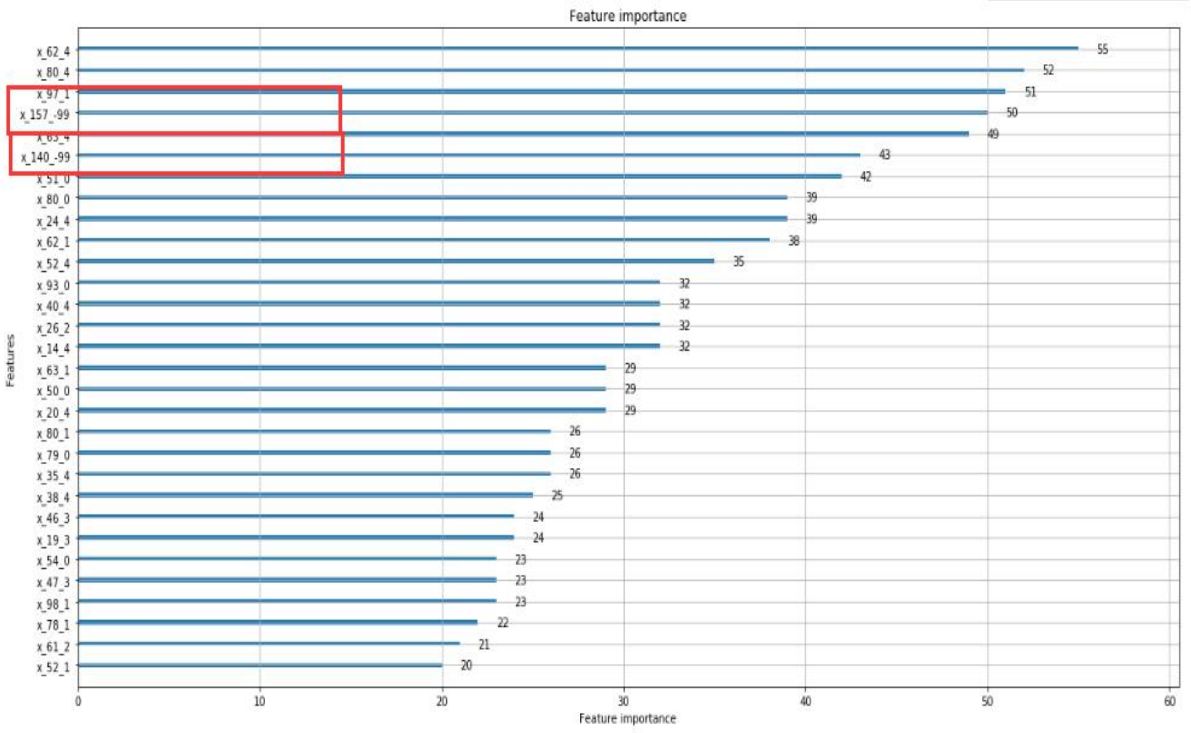


图 8：train set 离散化特征的重要性

因此，我决定不对类别特征中的缺失值做处理，**保留-99，代表是否为缺失值的特征**。实验对比发现，不处理的效果最好，也验证了上述观点。

三、特征工程

1.用户信息缺失值特征

前面针对每个 X 特征，在列方向上统计每个特征的缺失率，我们也可以针对每个样本（用户），在**行方向**上，统计每个用户的特征缺失个数，代表**用户的信息缺失程度**。在征信领域，用户个人信息的完整性，一定程度上，可以反映用户的信用程度，那是不是信息越完整，信用度越高呢？一开始直接统计每个样本的属性缺失个数，作为一个数值特征，加入模型，发现效果不好，于是开始作图分析原因。

我对数据按行统计每个样本的属性缺失值个数，作出 train set 的散点图：

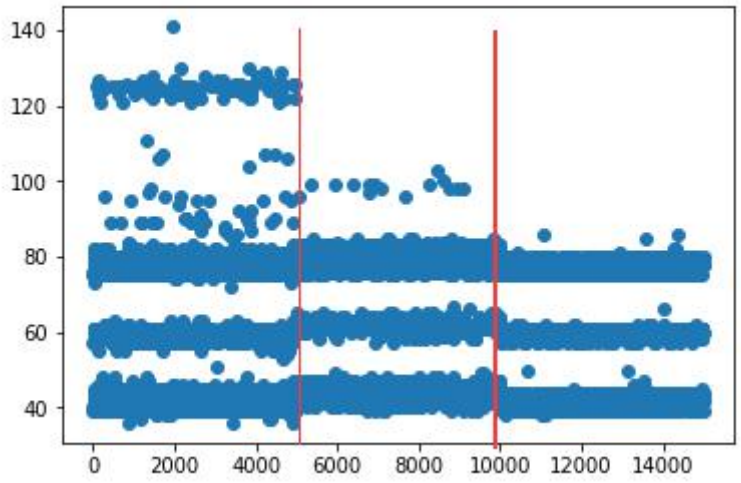
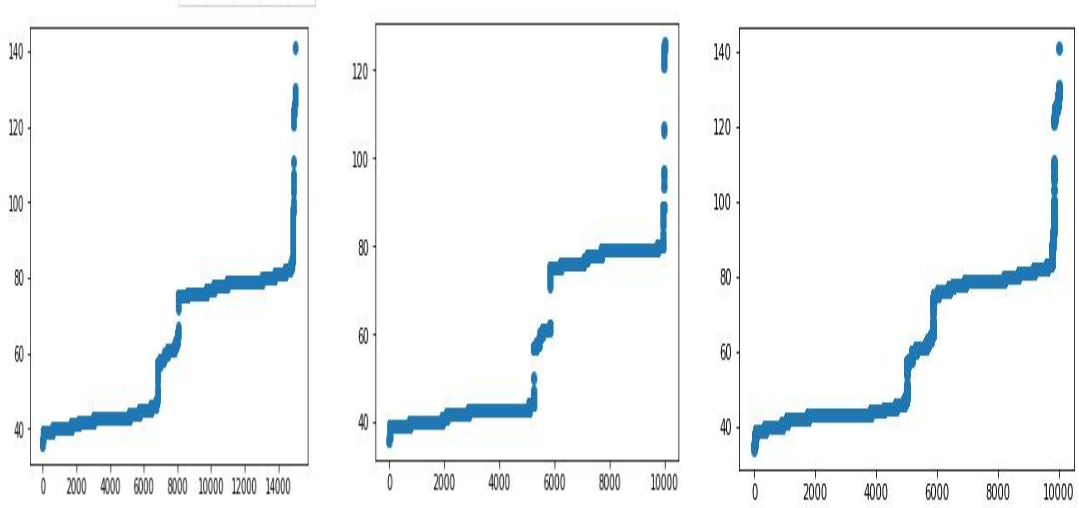


图 10：train set 样本属性缺失个数散点图

可以看出，对于不同的组别（group1:1-5000，group2:5001-10000，group3:10001-15000），样本的属性缺失数具有明显的层次性。而且观察数据会发现，train set 中的正样本（y=1），大多数会排在每个 group 的最后面，如图中红色分界线处，正样本的属性缺失值个数，具有一定的**层次性，分布在不同的区间内**，而并不是缺失数越大，越是信用差。

为了更细致观察。对缺失个数按照大小排序，作图如下：



(a) train set　　(b) test set　　(c) train_x set

对比三个数据集的样本发现，样本属性的缺失个数呈现**阶梯状**，三个数据集大体一致。

再根据样本的属性缺失个数的大小顺序，对 train set 中的负样本（y=0）进行累积和计算，画图如下：

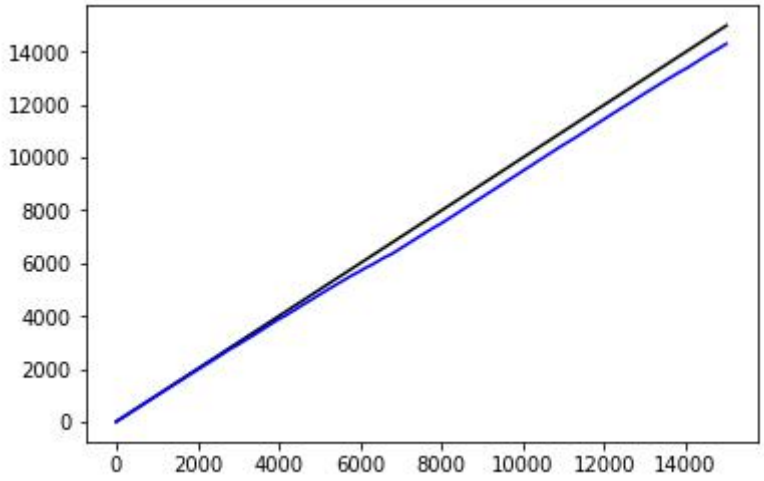


图 11：train set 中负样本随缺失数增长的累积和

综上所看，用户信用的好坏与用户属性的缺失个数并不是简单的数值线性关系，并不是说，信息缺失越严重，信用就越差——而是**信息缺失程度在某些特定区间，会出现信用差的样本（即 y=1）**。

所以，我们根据属性缺失个数的阶梯状信息，将缺失值个数进行离散化为 7 个区间（考虑到最大约为 140），从而将**数值特征离散化为等值的 7 个 bins**，得到 7 维度的缺失区间特征，线上线下同步提升 0.002 个点。

2.类别特征 one-hot 编码

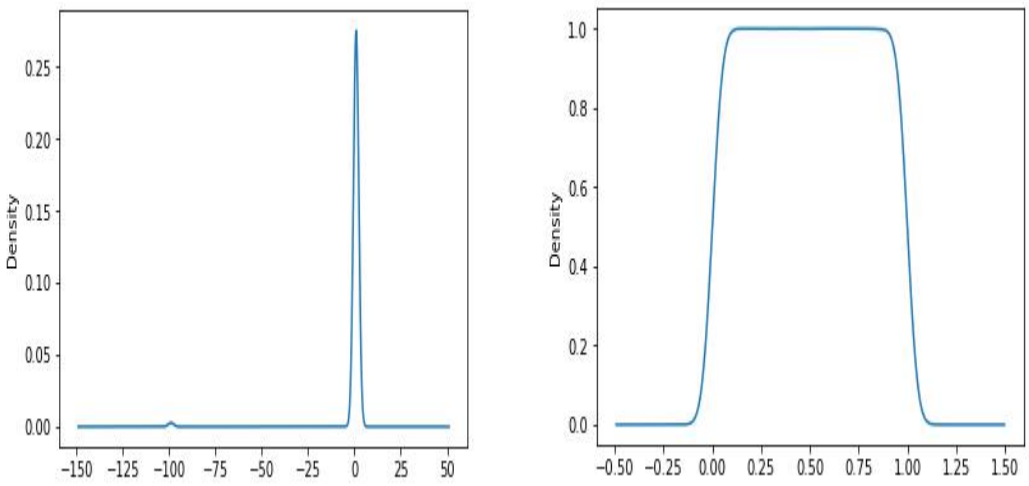
对于类别特征，一般模型（LR,SVM...）的做法都会 one-hot 编码，得到 01 特征，但是对于 lgb,xgb 这些集成树模型，是可以不用 one-hot 的。实验中，对于 lgb 模型，我对比进行和不进行 one-hot 的效果发现，保持原始类别方式效果稍好。在 lgb 的官方文档中，也有相应的介绍—— one-hot coding 对于学习树来说不是个好的解决方案，因为对于一个基数较大的类别特征，lgb 学习树会生长的非常不平衡，并且需要非常深的深度才能来达到较好的准确率。

而且，我们分析发现，大多数类别特征，重要性等于 0，说明进行 onehot 编码之后，会使得特征变得稀疏，会影响模型效果。再者，因为是匿名特征，我们不清楚特征含义，假如对于类似于“收入等级”这类型的类别特征，原始数据的类别数 1,2,3,5，包含了**等级的大小关系**，但是 one-hot 之后，却丢失了这种相对大小关系。

而对于 xgb 模型，我基于上面的考虑，只对 x_157,x_140 等非常重要的类别特征进行 one-hot，对比发现只有 0.0005 的波动提升。

3.数值 rank 特征

因为前面只对重要的数值特征进行均值填充，但是，还有很多的数值特征并没有进行处理，而且缺失值默认为-99，但是很多数值特征的取值大致范围在 0-5，这就造成**数据密度分布不均匀**，所以对其进行 rank 排序，并且归一化，如下图对比：



(a)原始数据密度图　　(b)rank 并归一化之后的密度图

对于树模型来说，数值类型特征是不需要进行归一化处理的，因为他并不关心变量的具体数值，而是变量的分布。这里对数值进行 rank，主要是为了保证模型**对异常数据的鲁棒性**，使其更加稳定，降低过拟合风险。实验表明，进行 rank 之后的特征，线上线下同步提升 0.0008。

4.其他尝试

下面是一些数值组合特征和类别特征构造的尝试，当筛选表现良好的特征，加入到原始特征后，线下有稳定的 0.01 的提升，但是线上却降低，出现明显的过拟合。包括后面尝试使用半监督，出现类似的现象，这里初步分析，应该是数据太少的原因，因为 A 榜才 40%的数据（4000 样本）。

①数值特征组合

因为是匿名特征，所以对所有的数值特征，两两进行数值计算，如 x*y,x/y,y/x,x+y,x-y,x*x + y*y 等，产生**交叉组合特征**，并通过树模型选择 topK 的特征，加入到原始特征。

②类别特征构造

针对数值特征，先对特征进行排序，然后再进行等量划分，相当于将用户按照数值排序进行**划分等级，对数值进行离散化**。并通过树模型选择 topK 的特征，加入到原始特征。

③是否缺失特征

针对 X 特征，构造该特征取值**是否为缺失值**，是则为 1，否则为 0，共得到 157 维度特征。

④深度特征

利用深度学习自编码技术和梯度提升树模型，训练所有数据，学习数据特征和内在联系，**自动生成深度特征**。并通过树模型选择 topK 的特征，加入到原始特征。

四、特征选择

常见的特征选择方法有：

①过滤式选择：通过相关系数、卡方检验、信息增益等筛选特征。

②包裹式选择：通过迭代特征，利用学习器的性能评估进行选择。

③嵌入式选择：利用学习器自动选择特征，包括，正则化、基于树模型选择。

本赛题，我主要采用③中的**基于树模型**进行特征选择，首先训练所有特征，根据模型得到的特征重要性排序，选择 topK 的特征进行训练和分析。而对于前面提到的重要特征，我又使用②中的**迭代特征的方式**，线下采用 5 折交叉验证，来判断一个特征或者特征子集是否加入。迭代特征选择方法如下：

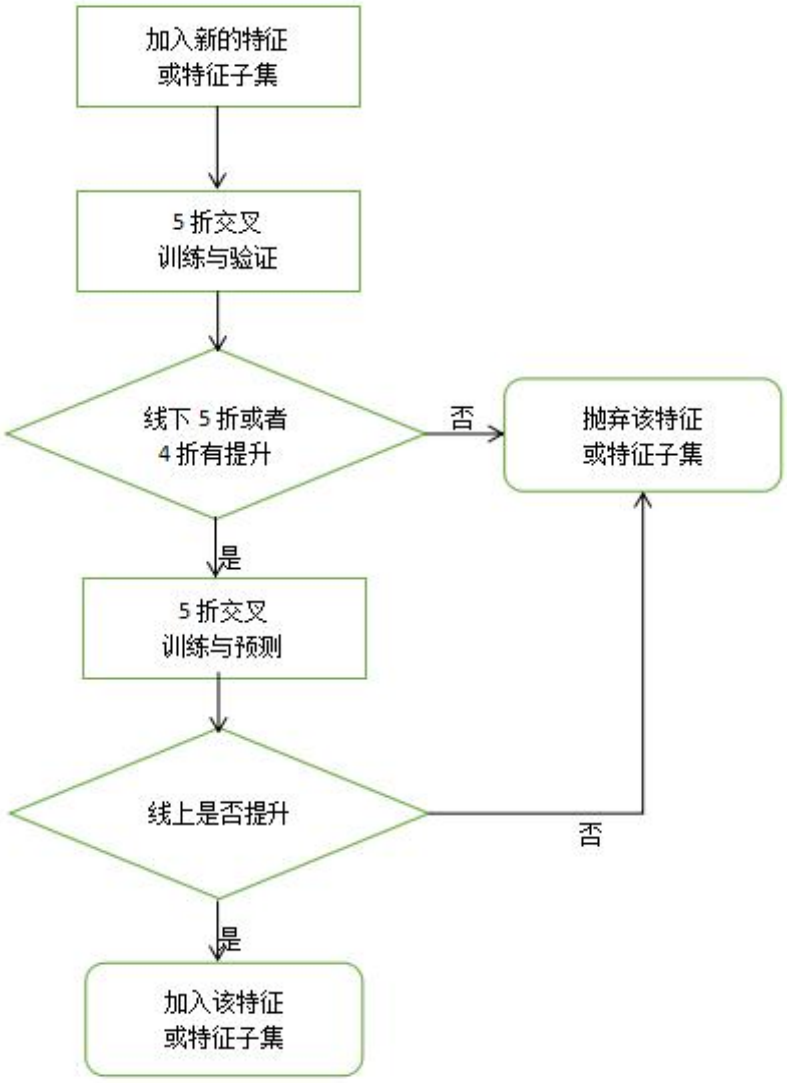


图 12：迭代特征选择方法

五、模型训练与评估

1.训练集/验证集划分

由于线上提交次数有限，为了快速验证方案，一般需要划分线下训练集、验证集。常见的做法是 train_test_split，随机选取 80%的数据作为 train set，剩下 20%的数据作为 validation set。由于本赛题**正负样本分布极其不均衡**，0-1 的比例竟然高达 21:1，随机划分会改变数据分布，而且无法做到有效验证，所以这里我采用，**5 折分层采样的方式（StratifiedKFold）**，进行数据划分，示意图如下：

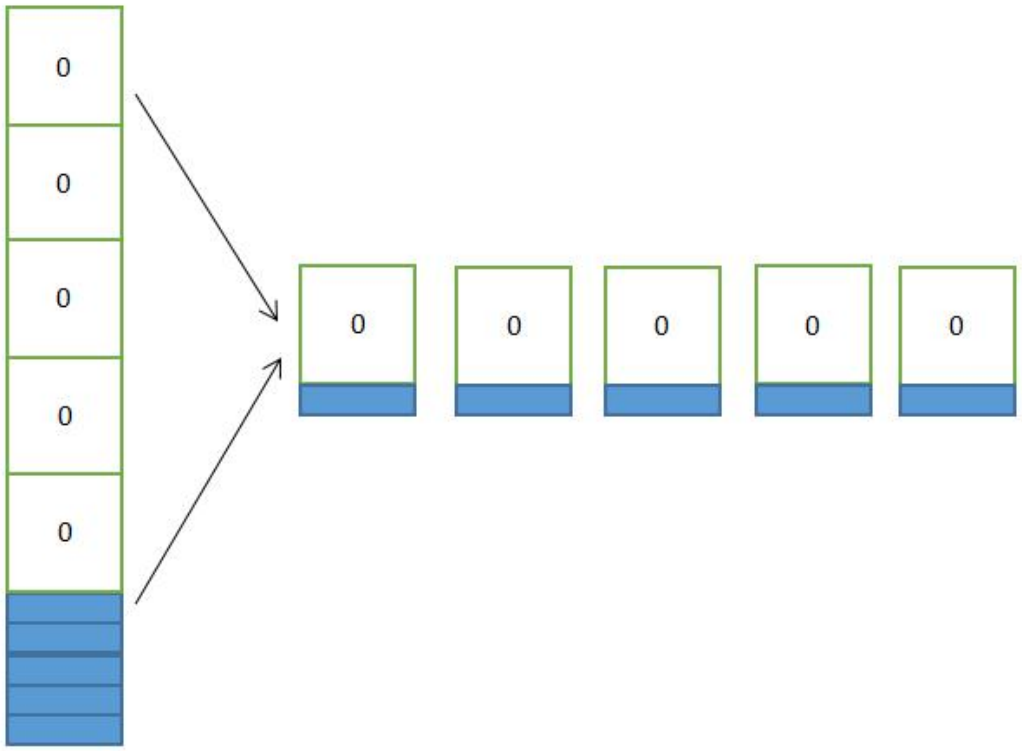


图 13：5 折分层采样示意图

在分析数据过程中，我们发现，group1 中的正样本比其他两个 group 要多，如下图。

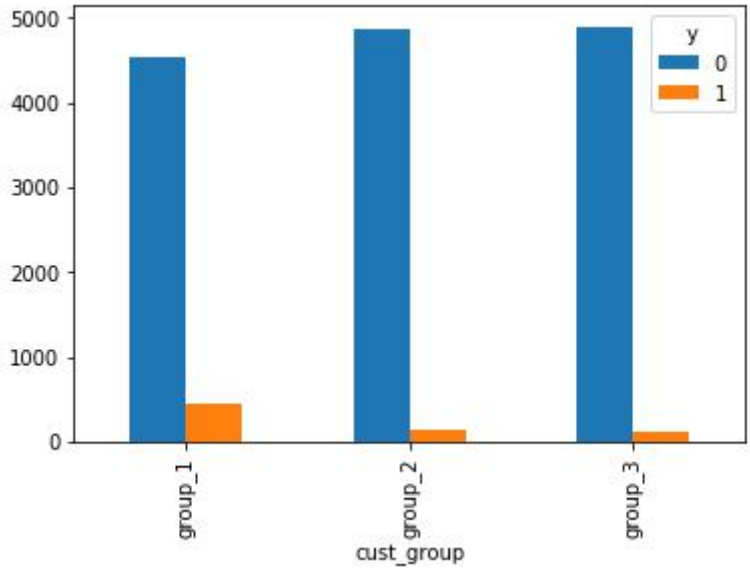


图 14：不同分组的正负样本比例图

针对这个问题，所以我也尝试对不用的分组，使用上述 5 折分层采样，然后再组合到一起，希望保持不同分组的正负样本比例，但是发现与直接做一个 5 折分层的效果没有太大区别，所以最后将不同组别的**数据 shuffle 打乱**，只做一次 5 折分层采样，这样能够使模型**对不同分组数据具有更好的泛化性能**。

2.五折交叉验证与融合

因为随机划分训练集和验证集，具有一定的随机性，不能确保能准确验证线上结果，为了模型更加稳定，我采用 5 折交叉验证的方式，进行线下模型验证，示意图如下。

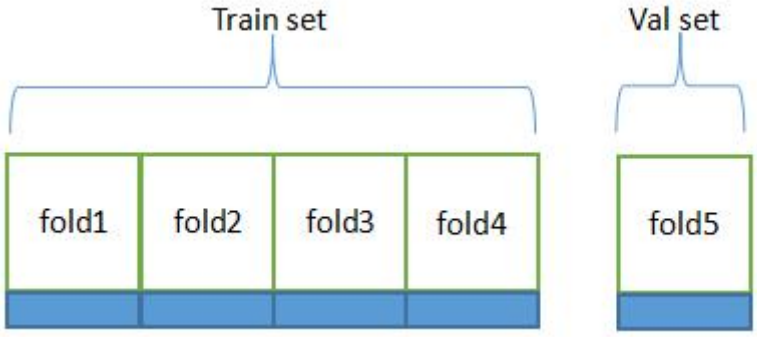
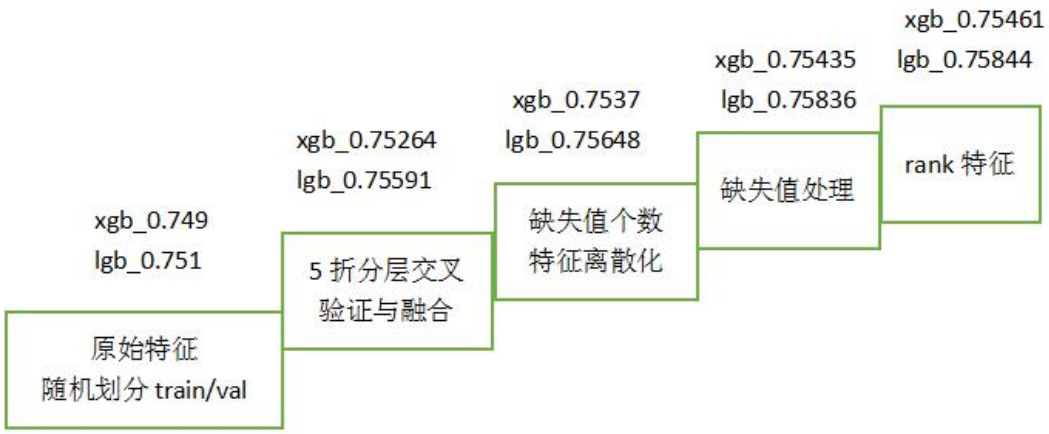


图 15：5 折交叉验证示意图

每次选取其中 4 个 fold 作为 train set，剩下 1 个 fold 作为 validation set，并且每折训练都采用早停（early-stoping），使用训练好的最佳模型进行 test set 的预测，最后对这 5 个不同数据分布的训练子集的预测结果，直接采用**结果平均融合**，有很大的性能提升（提升 0.003），并且能够保证模型的**稳定性**。

3.模型选择

我们通过对 lr(逻辑回归)、svm(支持向量机)，gbdt（梯度提升树）,rf(随机森林)，dnn（神经网络），lgb（LightGBM），xgb(XGBoost)等模型的对比，发现 lgb、xgb 模型的预测效果最好。其中，两个模型的具体表现如下：



A 榜	B 榜
Xgb_0.75461	0.75493
Lgb_0.75844	0.75478

图 16：模型表现图

最后**多个模型的结果融合效果并不好**，而且 stacking **很容易过拟合**，分析可能是数据过小的原因，A 榜的评测不能得到有效的反馈，所以，为了**模型稳定，防止炸榜**，最后分别提交 lgb、xgb 单模型的最好成绩。

4.加权 AUC/平均 AUC

线下的评测指标与线上保持一致，都是选取的 AUC。我尝试过：

①先对不同分组的 auc 进行加权求和，再对 5 折的 auc 取均值，线下平均 auc 基本维持在 0.79-0.81；

②不根据分组，直接对所有数据样本计算 auc，最后对 5 折计算平均 auc，基本维持为 0.81-0.82。

其中，为了反映某个特征的对于模型的性能提升与否，不能只单单看最后 5 折的平均 auc 是否提升，而是确保，**每一折或者至少其中 4 折都要有提升**，这样才能确保加入的特征不会影响模型的稳定性。

六、尝试与思考

1.迭代半监督

因为提供的数据含有很多无标签的样本（10000 个），所以我们可以采用半监督学习方法，提升模型性能。加上这是一个正负样本极其不均衡的问题，我们可以使用**半监督方法**，进行类似于正样本的**过采样**（over sample），常见的过采样方法有 SMOTE 和改进的 TSMOTE，但是这里既然提供了真实的样本数据，我们就不需要进行过采样——我们可以直接使用最好的监督模型，对无标签数据 train_x 进行预测，选取概率最大的 topK 个样本作为正样本，概率最小的 lowN 个样本作为负样本，保持 0-1 比例为（0：K，1：5---1:20），依次放入原来的 train set 中，看看模型对于 val set 的性能是否提升，有的话，则加入，并且预测 test set；否则不加入，重新选取，直到模型性能不提升为止。

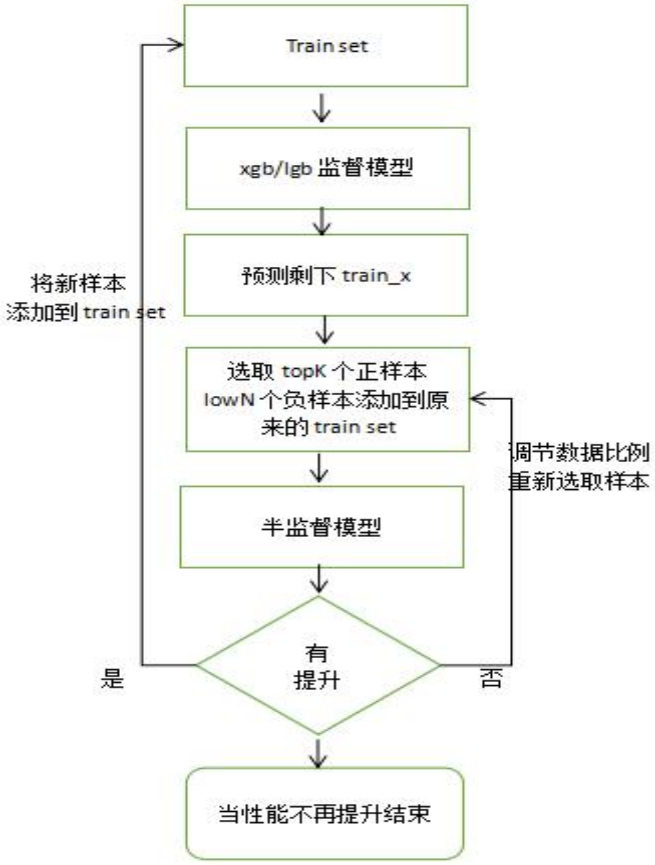


图 17：迭代半监督流程图

通过上面的方法，线下有接近 0.01 的提升，但是线上却出现过拟合的现象，我认为可能的原因有：①数据过小，容易过拟合 ②模型本身性能不理想，增加训练数据是没有效果的。但是我认为，1 的可能性更大。

2.特征组合与构造

在特征工程中介绍过，我构造了一些组合特征，线下都有稳定的提升，但是线上却也是过拟合。

3.过拟合现象的思考

我尝试过进行特征构造、半监督学习、过采样、模型融合等操作，都出现了过拟合现象——线下涨分，线上降分。我认为这是数据小造成的，因为测试数据才 10000，A 榜 40%，B 榜 60%，还是随机划分，所以具有很大的波动性。

当我意识到这点的时候，我可以肯定，这题需要一个稳定的模型。所以我采用 5 折交叉验证和融合的方式，对数据进行严格的分析，从而确保线上线下的一致性。最后提交的 xgb 单模型，A 榜 0.75461，B 榜 0.75497，只有万分位的波动。

结果切换 A/B 榜的时候，发现 A 榜的 top10 几乎全部过拟合严重，B 榜排名下降 20 多名，只有我从 top3 下降到 top9,而 B 榜的 top10,几乎人均排名上升 20 多，正好验证了我的猜想。

七、创新点与总结

1.数据可视化分析

通过对数据进行**可视化分析**，对不同数据属性的特征，进行不同的操作，有利于数据处理，做到知其然，知其所以然。

2.对特征缺失进行多维度分析

从**列方向上**，统计每个特征的缺失率，分析特征的重要性；从**行方向上**，统计每个用户样本的属性缺失值，分析用户的信息完整程度；并且对不用类型的特征，进行不同的缺失值处理，从而保证模型的性能提升。

3.数值特征 rank 化

对数值特征进行 rank，在归一化，可以减少缺失值-99 对模型的干扰，增加模型的**鲁棒性**。

4.五折分层、交叉验证与融合

考虑到数据的正负样本不均衡，以及确保模型**稳定性**，采用 5 折分层交叉验

证的方式，达到线下线上同步提升，最后采用 5 折结果平均融合，提升模型性能。

5.模型简单而稳定

首先，我们的模型简单，特征维度才 170 多维，都是**单模型**，运行时间几分钟，**效率高**。而且为了确保模型的稳定性，使用包裹式特征选择，每加一个特征或者特征子集，都确保 5 个 fold 的 val set 都有提升，或者至少 4 个，而不是单单只用 auc 的均值作为性能指标。正是因为这点，我的 xgb 模型，在 A/B 榜测试子集的成绩，只有**万分位波动**（0.75461----0.75493）