西南财经大学

Southwestern University of Finance and Economics

课程论文

学年学期：2020-2022 年第 1 学期

课程名称：机器学习

论文题目：银行借贷业务风险预测实例分析

学号：41809055、41809061、41809062、41809050

学生姓名：简义瀚、谢瀚洋、郑岳、马雪岩

学院：统计学院

年级专业：2018 级经济统计学

银行借贷业务风险预测实例分析

简义瀚 谢瀚洋 郑岳 马雪岩

Bank lending business risk forecast example analysis

摘要：本文以对比赛数据与模型的探索过程为思路，逐步对数据进行处理并给出最终预测。主要对数据进行了相关性分析，缺失值分析，特征提取，特征交叉。建立 *XGBoost* 模型和 *XGBoost* 模型对测试集进行预测，效果还算不错。为处理类似的风险预测问题提供一定的解决思路。

关键词：机器学习，*XGBoost*，*XGBoost*，贝叶斯调参，借贷风险预测，特征交叉，特征提取

一、引言

当下,互联网金融经过蓬勃发展,已呈现出多种多样的业务模式和运行机制。但互联网金融发展的同时也引发了信用风险和用户欺诈等问题。P2P 网贷作为互联网金融的突出代表,其所面临的信用风险尤为突出,故急需通过建立信用评分体系预测借款人逾期/违约倾向从而提高 P2P 网贷对其信用风险的控制水平[1]。这对未来互联网金融业可持续健康发展也具有重大意义。

然而,现实生活中这些天然带有多重数据源、超高维、稀疏等特点的复杂性数据也远远超出了线性回归或 Logistic 回归等线性模型所能处理的能力范围,这对传统风控提出了巨大的挑战。随着个人信息和各种行为数据的逐步完善,采用大数据挖掘技术预测个人未来的信用表现日益成为主流方法。如何在充分利用大数据的同时提高风控水平,正是传统风控转型为大数据风控的关键。

二、背景介绍

四川新网银行以建设“数字普惠银行”为愿景，运用互联网大数据风控、云计算、人工智能等技术，为客户提供具有高可得性和良好用户体验的金融产品。金融机构发放贷款后，需要对客户进行持续的风险跟踪和监测。综合利用客户的信用数据、行为数据等信息建立高风险客户识别模型可帮助金融机构及时发现风险并减少损失，因此，如何精准识别高风险客户是金融机构风险管理关注的重要问题。比赛邀请参赛者基于客户的基本信息和行为数据信息运用统计、机器学习 算法等工具建立模型，识别高风险客户。

三、数据介绍与解析

3.1、数据描述：

根据赛题任务，此次竞赛提供的数据包括用户 *id*，89 项脱敏的属性/行为特征，以及是否属高风险用户的标签项。可供下载和使用的有 5 个文件：

1.*y_train.csv*，训练集标签，包含和风险标签 ，共 15280 条

2.*data_b_train.csv*，训练集数据,包含用户基础信息，共 15280 条

3.*data_b_test.csv*，测试集数据，字段同 *data_b_train.csv*，共 5767 条

4.*data_m_test.csv*，测试集数据，字段同 *data_m_train.csv*，共 75255 条

5.*data_m_train.csv*，训练集数据，包含用户行为信息，共 173100 条

3.2、变量维度描述:

1.*data_b_train* 是客户基本信息表，是脱敏的客户人口统计和基本情况信息，变量主要包含性别、年龄、教育背景及其他申请时客户授权采集的各类画像标签。

2.*data_m_train* 是截至某个观察时间点的客户行为信息表，是脱敏的客户贷款发放后的行为数据信息，变量主要包含贷款后各个时间点上的提还款行为、逾期行为及其他各类行为画像标签。

3.*y_train* 是客户的表现标签信息，是在观察时间点后的风险表现，标签变量名称为 *"target"*，0 代表低风险客户，1 代表高风险客户 。

注：

1.特征变量名称以 *"x_"* 开头， 其中，以 *x_num_* 开头的是数值型变量，以 *x_cat_* 开头的是类别型变量。

2. 行为信息表中，*Timestamp*是时间标识，*Timestamp*越大离观察时间点 越近。

3.3、赛题解析

依据比赛主办方给出的相关描述，赛题内容属于金融领域的风险控制场景， 而该领域通常存在高维数据、稀疏数据、类不均衡等固有的数据问题。而比赛所提供的数据为真实业务场景下的脱敏数据，特征字段全部匿名化，因此无法使用金融领域的先验知识进行特征选择。只能通过对数据的观察进行数据的处理和特征的筛选与提取。

数据来自多产品（客群），因此，数据分布不一致问题较为突出。同时提供到用户的高维特征数据和面板数据（部分截面数据，部分面板数据），对应到现实场景中属于同时利用静态用户属性和动态的行为数据，预测用户是否风险较高。该问题实质是一个类不均衡的二分类问题，*0:1* 比例约为 *9:1*，评估指标为综合使用精确度（*Precision*）和召回率（*Recall*）的一个指标，并提供有标签的训练集样本，能够使用相关算法综合预测用户是否存在高风险。

四、数据预处理

4.1 缺失值分析

赛题没有给出明确的缺失值表示形式，使用 *pandas* 中的 *isnull 方法*进行探测，发现各列并无缺失值。但在通过 Excel 等方式快速浏览数据后发现，大部分 数据取值范围在[0,1]之间，而仅有少量数据出现-99 的异常值，使得存在-99 的特征方差会比其他特征更大。于是将*-99* 当作缺失值，对数据进行更新。更新后对每一个数据集单独进行缺失值分析。

4.1.1*data_m_*

data_m_train 一共有 *173100* 行，*68* 个特征，包括用户 *id*，行为时间以及 66 个行为描述数据，除了 *id* 和行为时间为整数类型数据，其余全为浮点数类型数据。

运用 *pandas* 中的 *info* 方法对数据进行观察，发现特征*'x_num_19'*、 *'x_num_38'*、*'x_num_39'*、*'x_num_10'*和*'x_num_27'*存在缺失值，其中*'x_num_19'*和 *'x_num_38'*全部为缺失值；*'x_num_39'* 有 *8199* 个缺失值； *'x_num_10'*和*'x_num_27'*的缺失值个数相同，全为 *107896* 个。

表 1 *data_m_train* 缺失值情况

特征	缺失值个数
'x_num_10'	*107896*
'x_num_19'	*173100*
'x_num_27'	*107896*
'x_num_38'	*173100*
'x_num_39'	*8199*

由于*'x_num_10'*和*'x_num_27'*的缺失值个数相同，猜想这两个特征可能有某些特殊的联系，将这两列单独提取出来进行比较，发现两列信息完全重合。如果同时加入模型会导致信息被重复使用，造成信息冗余，所以决定选取其中之一进行删除。

*而'x_num_19'和 'x_num_38'*全部为缺失值，对模型训练没有提供任何有效信息，所以将这两列也作删除处理。

综上将*'x_num_19'*、 *'x_num_38'*、*x_num_27'*这三列删除。

相对应的也将 data_m_test 中的'x_num_19'、 *'x_num_38'*、 *x_num_27'*这三列删除，保持训练集和测试集的结构相同。

4.1.2*data_b_*

data_b_train 一共有 *15280* 行，*19* 个特征，包括用户 *id* 以及 *18* 个基本信息数据。

运用 *pandas* 中的 *info* 方法对数据进行观察，发现特征*'x_num_1'*、 *'x_num_2'*和 *'x_num_3'* 存在缺失值，其中*'x_num_2'*有 *639* 个缺失值； *'x_num_1'*和*'x_num_3'*的缺失值个数相同，全为 *263* 个。

表 2 *data_b_train* 缺失值情况

特征	缺失值个数
'x_num_1'	*263*
'x_num_2'	*639*
'x_num_3'	*263*

由于*'x_num_1'和'x_num_3'*的缺失值个数相同，猜想这两个特征可能有某些特殊的联系，将这两列单独提取出来进行比较，发现两列具有缺失值的索引一致，但是非缺失值位置对应的数并不相同。于是考虑将这两列都保存。

4.1.3 缺失值填充与否

在比赛开始时，考虑建立随机森林模型进行预测，随机森林是一个包含多个决策树的分类器，对于存在缺失值的数据无法进行训练，于是对缺失值进行填充。缺失值填充的一般方法有均值填充，众数填充，中位数填充，以及随机森林填充。对以上的四种填充方式都予以尝试后发现提交结果并不好，于是后转为使用更加强大的 *XGboost* 模型和 *LightGBM* 模型，这两种模型可以对含缺失值数据进行拟合预测，所以不对缺失值进行填充。

4.1.4 缺失值隐藏信息提取

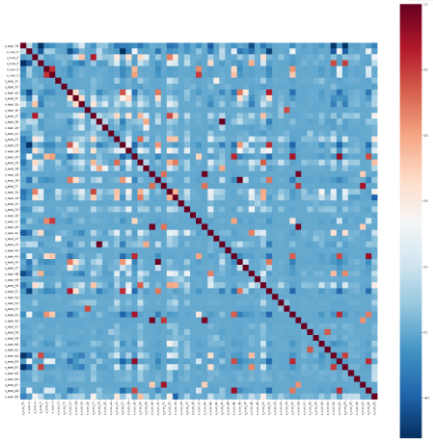
如 *data_*中的*'x_num_1'和'x_num_3*两个特征，缺失值索引一致，但是非缺失位置值不一致，说明缺失值可能隐含某种信息，于是根据特征值是否为缺失来构造新的 *0,1* 分类特征。*data_m_* 中对 *'x_num_10'和'x_num_39*进行是否缺失指标构建，指标命名为*'x_num_10_nan'和'x_num_39_nan'*；*data_b_*中对*'x_num_1'和'x_num_2*进行是否缺失指标构建，指标命名为*'x_num_1_nan'和'x_num_2_nan'*。

4.2 特征相关性分析

相关分析是指对两个或多个具备相关性的变量元素进行分析，从而衡量两个因素的的相关密切程度。经济变量中之间存在相关性很是常见，而对于本题中的脱敏数据，也需要对特征进行相关性探究，若是特征间的相关性很强，说明特征的信息重叠较为严重，直接代入模型进行分析很可能造成较大的误差，所以需要对相关性很强的特征进行处理。

4.2.1*data_m_train*

利用 *pandas* 中的 *corr* 方法得到特征之间的相关性矩阵，用 *seaborn* 库画热力图如下。



格子颜色越深代表相关性越强，而图中有少数深红的格子，其余大都为浅色，可以得到有少数的特征之间的相关性较强，但是由于特征数太多，直观通过热力图找到具有强相关关系的特征组合不方便，所以进行进一步探究，得到相关系数超过 *95%*的有以下特征组合，如表 3。

表 3 data_m_train 相关系数表（超过 95%）

特征 1	特征 2	相关系数
'x_num_18'	*'x_num_43'*	0.9998
'x_num_29'	*'x_num_40*	0.9891
'x_num_29'	*'x_num_56'*	0.9793
'x_num_30	*'x_num_46*	0.9999
'x_num_40'	*'x_num_56'*	0.9690

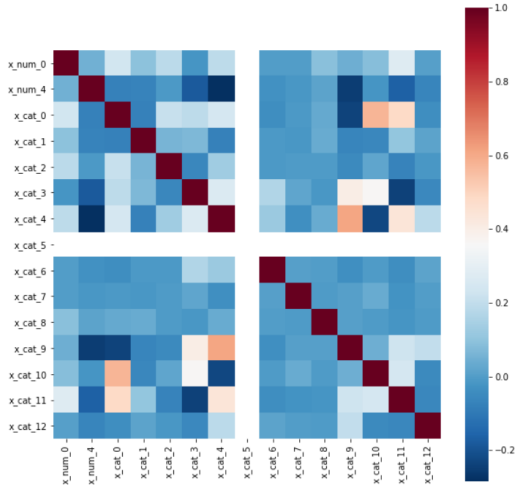
相关系数超过 95%，可以说明两者之间信息重合程度过高，甚至可以做近似的替代，于是对表 3 中特征组合各取一个特征进行删除，由于*'x_num_29'*、*'x_num_40*、*'x_num_56'*三者之间的相关系数都很高，于是保留一个，其余两个特征删除。

综上将*'x_num_40*、*'x_num_56'*、*'x_num_43'*、*'x_num_46*这四个特征删掉。

为保持训练集与测试集结构一致，将*data_m_test*中的这四个特征同步删除。

4.2.2*data_b_train*

利用 *pandas* 中的 *corr* 方法得到特征之间的相关性矩阵，用 *seaborn* 库画热力图如下。



格子颜色越深代表相关性越强，而图中除了处于对角线上的格子外，并没有深红的格子，大都为浅色格子，只有少数几个特征组合相关系数超过 *0.6*，但是最高也不超过 *0.8*，说明 *data_b_train* 中各特征相关系数都不高。但是 *'x_cat_5'* 这一特征与其他特征的相关系数格子全为白色，对其进行探究发现这一特征全为 *0* 值，并没有提供有效信息，于是将这一特征做删除处理。

为保持训练集与测试集结构一致，将 *data_b_test* 中的 *'x_cat_5'* 特征同步删除。

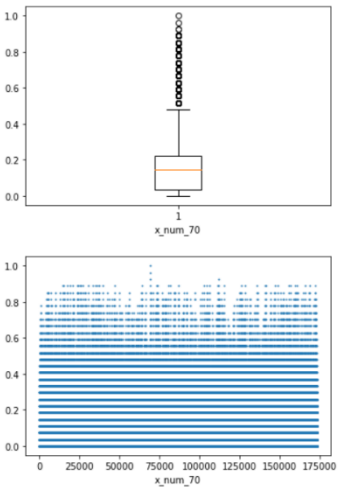
4.3 散点图箱线图观察

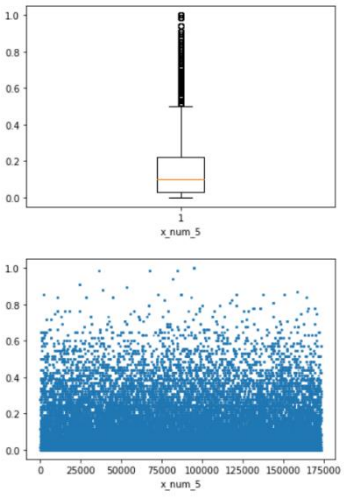
在数据分析比赛中，样本量较大时，通常对每一个特征画散点图和箱线图进行观察，以探究其分布特征以及是否有异常情况。

4.3.1*data_m_train*

用两种图（箱线图和以 *id* 为横坐标的

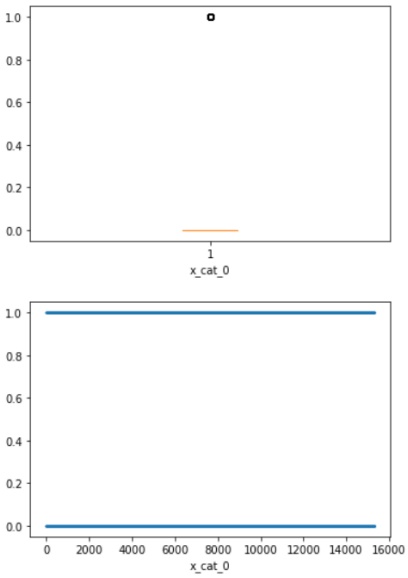
散点图）对各个特征数据进行呈现。发现 *data_m_train* 中的数据大致可以分为两个类型，一种是离散型数据，特征的取值为某些特定的数，散点图呈现较为规整，为条纹状，箱线图的异常值分布较为稀疏，如图 *‘x_num_70’*；另一种是连续性数据，取值较为杂乱，散点图和箱线图的异常值分布都较为密集, 如图 *‘x_num_5’*。





4.3.2*data_b_train*

用两种图（箱线图和以 id 为横坐标的散点图）对各个特征数据进行呈现。发现 *data_b_train* 中的数值型特征同样可以分为两个类型，离散型和连续性，如 *‘x_num_1’*。而类别型特征，取值为 0 或 1，散点图呈现较为规整，为两条线，箱线图为一点一线，如图 *‘x_cat_0’*。



4.4 *timestamp* 特征异常值处理

经观察得到行为信息的时间 *timestamp* 特征具有异常值，有些用户用 0 标记的行为数据出现两次，如下图。

	id	timestamp	x_num_70	x_num_5
1595	143	0	0.000000	0.014706
1596	143	0	0.000000	0.029412
1597	143	1	0.037037	0.029412

所以对 *timestamp* 特征异常值进行处理，具体思路如下：

1.找到 *timestamp* 特征具有异常值的 id

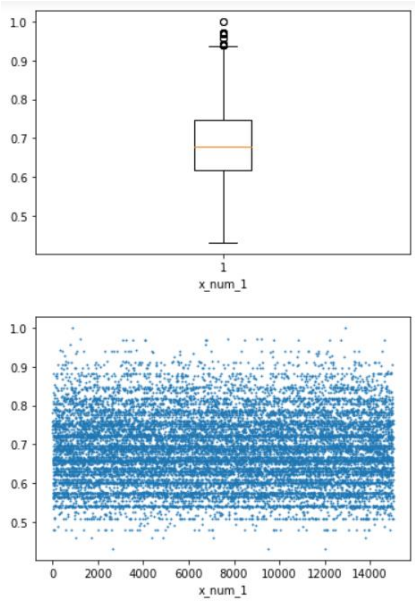
2.将该 *id* 所对应的数据单独提取出来

3.把 *timestamp* 中第一个 0 后面的 *timestamp* 都加上 1

五、特征工程

5.1 *Featuretools* 生成复杂时间序列衍生变量

Featuretools 是由 *Feature Labs* 公布的开源项目。它擅长于将时间和关系数据集转换为机器学习的特征矩阵。主要原理是针对多个数据表以及它们之间的关系，通过转换 *(Transformation)* 和聚合 *(Aggregation)* 操作自动生成新的特征。转换操作的对象是单一数据表的一列或多列（例如对某列取绝对值或者计算两列之差）；聚合操作的对象是具有父子 *(one-to-many)* 关系的两个数据表，通过对父表的某列进行归类(groupby)计算



子表某列对应的统计值。

5.1.1 Featuretools 包自动生成衍生变量

将 *data_m* 中每个用户对应的各特征指标集合到 *data_b* 中形成新的 *data_b*，这些指标包括数值型特征的最大值、最小值、平均值、总和、标准差、偏度以及分类型特征的众数。处理过后的新 *data_b* 含 385 个特征，生成特征的标签含义如下。

表 4 标签含义表

标签	MAX	MIN	MEAN	SUM	STD	SKEW	MODE
含义	最大值	最小值	平均值	总和	标准差	偏度	众数

5.1.2 自动生成特征的处理

因为 *Featuretools* 包自动生成衍生变量不仅根据数据的特征进行生成，所以新生成的特征很可能出现一些异常，主要为特征信息无效，即特征的值唯一和特征间相关系数过高，具有很多重复信息。

首先探究无效信息，将新 *data_b* 特征值唯一的特征提取出来予以删除。然后对特征之间的相关系数矩阵进行探究，由于特征数过多，不宜进行画图观察，所以写一个提取相关系数过高的特征组合的函数，将特征组合相关系数高于 95%的特征输出，并将具有重复信息特征组合留取一个后删除。

```
def get_corr_over_fea(corr_max):
    m_drop_list=[]
    fea_name=df_corr.columns
    for i in range(len(fea_name)):
        for j in range(i+1,len(fea_name)):
            corr=df_corr.iloc[i,j]
            if corr>corr_max:
                m_drop_list.append(fea_name[j])
                print('{}和{}相关系数为{}'.format(fea_name[i],fea_name[j],corr))
    return m_drop_list
```

```
df_corr=data_b_train_plus.corr(method='pearson')
b_drop_list=get_corr_over_fea(0.95)
```

```
x_num_0和MAX(data_m.x_num_51)相关系数为0.9794069358781137
x_num_0和MEAN(data_m.x_num_51)相关系数为0.9821600465061274
x_num_0和MIN(data_m.x_num_51)相关系数为0.9796062825438254
COUNT(data_m)和MAX(data_m.timestamp)相关系数为0.9580047491776158
COUNT(data_m)和STD(data_m.timestamp)相关系数为0.9506537151931251
COUNT(data_m)和SUM(data_m.timestamp)相关系数为0.9537159905578719
MAX(data_m.timestamp)和MEAN(data_m.timestamp)相关系数为0.9960005647074865
MAX(data_m.timestamp)和STD(data_m.timestamp)相关系数为0.9990616013274193
```

5.2 合并 *data_b* 与 *data_m*

由于用户信息分为两个数据表格，一个是行为信息 *data_m*，一个是基本信息 *data_b*，而用模型进行拟合预测时只能对单独一个表进行处理，所以需要将 *data_b* 与 *data_m* 拼接起来形成一个大表，这里采用 *pandas* 中的 *merge* 方法根据用户 *id* 将信息整合。

5.3 重要特征进行特征交叉

使用手动调参后最优的 XGBoost 模型对训练集进行训练，由于 XGBoost 模型的基学习器是树模型，所以可以得到模型训练中各个特征的重要性，选取其中最重要的 5 个特征进行特征交叉，即两两相乘得到 10 个新的特征，经过检验得到，这 10 个特征的加入对预测结果有所改善，即他们之间的乘积有一定的经济含义。

	0	1
78	MEAN(data_m.x_num_16)	0.011217
269	timestamp	0.012668
163	SKEW(data_m.x_num_21)	0.015998
113	MEAN(data_m.x_num_65)	0.027675
27	MAX(data_m.x_num_20)	0.029065

六、模型的选择与训练

6.1 XGBoost 算法介绍

XGBoost 是一种以 *CART* 树为基学习器的提升算法，是对 *GBDT*（*Gradient Boosting Decision Tree*）算法的一种改进。

所谓提升算法（*Boosting*），即是构建多个弱学习器对数据集进行预测，再以 某种

策略将弱学习器的预测结果结合起来，形成最终的预测结果，以实现弱学习 器到强学习器的提升[2][3] 4]。*XGBoost* 算法以决策树为弱学习器，总共构建 *T* 棵树，当构建到第*t* 棵树时，对前*t−1* 棵树训练样本分类回归产生的残差进行拟 合。拟合产生新的树时，遍历所有可能的树，并选择使得目标函数值最小的树。 通常，该步骤会分解进行，即构造新的树时，每次只产生一个分支，选择最好的 那个分支[5][6]。

该算法能够为模型带来高度的预测准确性与泛化性能。近年来，无论是在机器学习科研领域还是在数据挖掘竞赛中，*XGBoost*的出色表现都表明它是目前用于分类或是预测的最优秀的算法之一。

6.2 LightGBM 算法介绍

LightGBM 是个快速、分布式的、高性能的基于决策树算法的梯度提升框架。 可用于排序、分类、回归以及很多其他的机器学习任务中。

因为他是基于决策树算法的，它采用最优的 *Leaf-wise* 策略分裂叶子节点， 然而其它的提升算法分裂树一般采用的是 *Level-wise*。因此，在 *LightGBM* 算法中，当增长到相同的叶子节点，*Leaf-wise* 算法比 *Level-wise* 算法减少更多的 *Loss*， 因此导致更高的精度。与此同时，它的速度也让人感到震惊，这就是该算法名字 *Light* 的原因。*LightGBM* 在 *Leaf-wise* 之上增加了一个最大深度的限制，在保证高效率的同时防止过拟合。

6.3 贝叶斯调参介绍

贝叶斯优化用于机器学习调参由 J. Snoek(2012)提出，主要思想是给定优化的目标函数(广义的函数，只需指定输入和输出即可，无需知道内部结构以及数学性质)，通过不断地添加样本点来更新目标函数的后验分布(高斯过程,直到后验分布基本贴合于真实分布)。简单的说，就是考虑了上一次参数的信息，从而更好的调整当前的参数[7][8]。

贝叶斯调参与常规的网格搜索或者随机搜索的区别是：

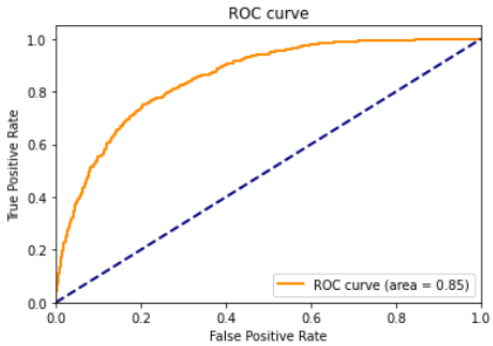
- 贝叶斯调参采用高斯过程，考虑之前的参数信息，不断地更新先验；网格搜索未考虑之前的参数信息[9]
- 贝叶斯调参迭代次数少，速度快；网格搜索速度慢,参数多时易导致维度爆炸
- 贝叶斯调参针对非凸问题依然稳健；网格搜索针对非凸问题易得到局部优最[10]

6.4 调参过程

6.4.1*XGBoost* 模型

基于测试集数据分别对 *XGBoost* 模型进行贝叶斯调参，得到表现最好的一组超参数如下左图，对这组参数进行验证，得到 ROC 曲线如右图，说明拟合效果不错。

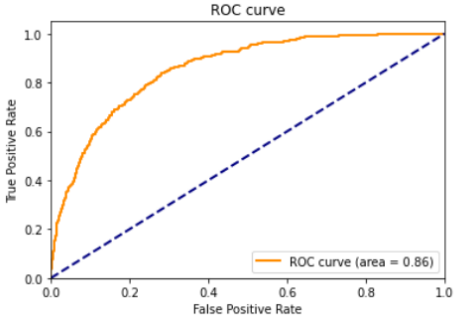
```
xgbmodel = XGBClassifier(
 learning_rate =0.01,
 n_estimators=951,
 max_depth=5,
 min_child_weight=1,
 gamma=0.356,
 subsample=1,
 colsample_bytree=0.98,
 alpha = 5.0,
 nthread=4,
 scale_pos_weight=1,
 seed=27)
```



6.4.2 *LightGBM* 模型

基于测试集数据分别对 *LightGBM* 模型进行贝叶斯调参，得到表现最好的一组超参数如下左图，对这组参数进行验证，得到 ROC 曲线如右图，说明拟合效果也不错。

```
params = {
    'task': 'train',
    'boosting_type': 'gbdt',  # 设置提升类型
    'objective': 'binary', # 目标函数
    'metric': 'binary_logloss',  # 评估函数
    'max_depth':4,
    'num_leaves': 102,   # 叶子节点数
    'learning_rate': 0.01,  # 学习速率
    'feature_fraction': 0.6, # 建树的特征选择比例
    'bagging_fraction': 0.95, # 建树的样本采样比例
    'bagging_freq': 5,  # k 意味着每 k 次迭代执行bagging
    'verbose': 0,
    'nthread': -1}
```



6.4.3 最终模型确定

基于测试集数据分别对 *XGBoost* 模型和 *LightGBM* 模型进行贝叶斯调参，得到两个模型下表现最好的一组超参数，用最优超参数下的 *XGBoost* 模型和最优超参数下的 *LightGBM* 模型分别对测试及数据进行预测，并将处理后的结果提交至平台，结果表明 *LightGBM* 模型表现更好。

七、思考与总结

由于第一次参加类似比赛，也是刚刚接触机器学习，知识储备并不足够，所以最终获得名次并不高。但是过程中我们做了很多的尝试，从决策树到随机森林，再到 *XGBoost* 和 *LightGBM*，模型在一步步改进，数据的处理以及特征工程上面的经验也慢慢积累，现在看来，我们还可以做更多的尝试，比如进行模型融合，使得模型不只限于同质性的基学习器，还可以使用一些深度学习算法进行预测等等。

机器学习这门课感觉十分有用，像是打开了一座数据分析的大门，我们之后也会多了解这个方向，努力学习。很感谢邓蔚老师和姜志豪学长一学期的陪伴，在此致谢。

八、参考文献

[1]牛丰，杨立．基于博弈理论的 P2P 借贷信用风险产生机制分析［J］．财务与金融，2016(1)

[2]梁云 1991，XGBoost 原理概述 XGBoost 和 GBDT 的区别，

http://www.elecfans.com/d/995278.html，2020-09-20

[3]Binbin Yang, Songqing Shen, and Wei Gao. “Weighted Oblique Decision Trees,” national conference on artificial intelligence,2019:5621-5627.

[4]Pritom Saha Akash, Md Eusha Kadir , Amin Ahsan Ali and Mohammad Shoyaib. “Inter-node Hellinger Distance based Decision Tree,” international joint conference on artificial intelligence,2019 : 1967-1973.

[5]Guolin Ke, Qi Meng , Thomas William Finley , Taifeng Wang , Wei Chen ,Weidong Ma, et al. “LightGBM: a highly efficient gradient boosting decision tree,” neural information processing systems,2017: 3149-3157.

[6]Vincent Grari, Boris Ruf , Sylvain Lamprier and Marcin Detyniecki. "Fair Adversarial Gradient Tree Boosting." international conference on data mining ,2019.

[7]Kim J，Lee D，Chung K Y． Item recommendation based on context-aware model for personalized u-healthcare service ［J］． Multimedia Tools and Applications，2011，71(2) : 855 － 872

[8]Shahriari B，Swersky K，Wang Z，et al． Taking the Human Out of the Loop: A Review of Bayesian Optimization［J］． Proceedings of the IEEE，2016，104(1) : 148 － 175．

[9]崔佳旭，杨博． 贝叶斯优化方法和应用综述［J］． 软件学报，2018，29(10) : 176 － 198．

[10]柴慧敏，赵昀瑶，方敏． 利用先验正态分布的贝叶斯网络参数学习［J］． 系统工程与电子技术，2018，40(10) : 219 － 224