# 基于借贷平台数据的用户行为分析

**19213225 涂庆**

摘要

面对越来越流行发展的金融借贷形势，借贷公司如何更好地降低贷款违约风险是一个值得关注的问题。本文讨论的问题是关于lendingclub信贷平台上的公开数据作为小微企业主信贷数据模拟样本，并用此建立合适的分类模型进行实际生产预测。

首先本文采用数据预处理和特征工程将数据进行清洗和加工，提高导入模型的数据质量，然后通过借贷信息特征分析、客户画像以及借贷业务分析三个角度的可视化直观分析数据中蕴含的信息；最后对比三种不同模型的表现效果和时间成本，推荐最优算法模型运用于实际业务开展和模型构建过程。

本文的优势在于，在原数据特征基础上，进行了独热编码等，提高了特征在算法中的解释性，利于算法更好的拟合真实情况；同时使用了专门针对高维稀疏特征以及样本不平衡的算法，进一步缓解了数据本身存在的劣势。论文所得算法模型适用于借贷平台风险监测领域，对借贷公司、俱乐部等机构更好降低借贷者信用风险具有一定的实际指导意义和价值。

**关键字：**不平衡样本、高维稀疏、随机森林、Xgboost、easy ensemble算法

1 问题重述

信用风险是金融监管机构重点关注的风险，关乎金融系统运行的稳定。在实际业务开展和模型构建过程中，面临着高维特征、属性信息交叉等各种问题，如何应用机器学习方法等数据挖掘方法提高信用风险的评估和预测能力，是各种类型的金融机构积极探索的方向。

论文使用的数据集的对象是关于借贷平台公司，其中包含的属性特征与国内的借贷平台公司类似，也是作为个人对个人的一种借款平台，主要做的是撮合业务。在一端募集有投资需求的资金，给予一定投资收益。另一端则将资金已贷款形式发放于有资金需求的客户，收取一定息费。目前国内排名前三的平台为陆金服，玖富普惠，宜人贷。由于承诺投资收益的确定，该业务的盈利核心即控制贷款端的风险，降低坏账。而率贷款的风险把控，无论国内国外，方法都是相近的，需要的字段也就相近，因此数据集的差异性也较小。

论文从数据集提供实际业务场景中的借贷数据作为建模的对象，所采用的数据集内容主要分为两个方面：一为借款申请时的基本信息，即贷前信息；二为还款信息，记录还款情况及逾期情况，为贷后信息。希望可以能借此运用所学的机器学习算法和数据挖掘的知识来锻炼和检验实战能力，同时也希望论文的模型结果对于相关借贷平台机构更好的保证公司系统运行的稳定性提供一定的实际指导意义。

2 数据预处理与特征工程

2.1 数据集描述

论文所使用的数据只有一个数据表（load.csv，大小为22MB），数据表内容：共有39786 行用户记录，其中有55个特征属性，由此可见数据集是属于高维特征属性数据；其中属性内容可以划分为两块内容：

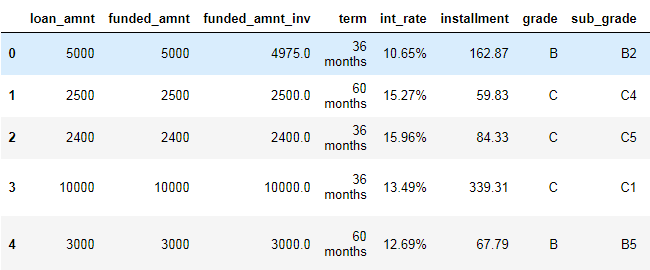
（1）借款申请时的基本信息，即贷前信息。

其中又分为两块：

1. 申请借款的基本情况：借款金额，借款期限，借款利息等属性信息；

2. 申请人的基本信息：借款人风险等级，工作年限，住房性质，年收入，收入来源，借款状态，借款用途，负债率，征信情况等属性信息；

（2）还款信息，记录还款情况及逾期情况，为贷后信息。

下面展示部分属性信息和内容如下：

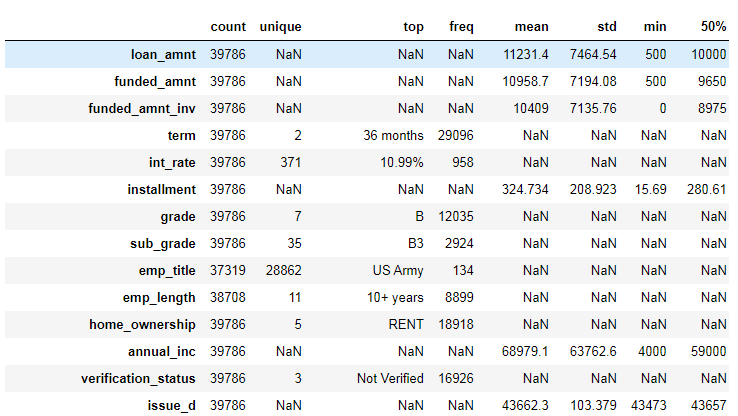
论文的目的是要预测借贷人是否违约还款，下面计算数据集中还款情况，得到正负标签之比：

**(Y=0): (Y =1)= 6.09913947651488:1**

由此可知，该二分类问题面对的是不平衡数据对象，同时存在高维特征属性，这两点是影响论文数据处理和算法选择的关键因素。

下面针对论文采用的数据进行数据预处理和特征工程，进一步提升数据的质量，提升后续数据挖掘和模型的效果。

2.2 数据预处理

根据上述简单的数据集描述，现在对数据集进行描述性统计展示（只展示部分属性）：

由描述性统计结果可以看出，数据集中很多属性存在缺失值的情况。因此对采用的数据集有一定初步认识，现在论文针对每个特征属性，进行详细的数据探索分析。

2.2.1 重复值查找

由于数据量大和高维特征，考虑查找数据集中是否存在大量的重复值，常见大量重复值出现的原因有：重复记录。论文采用数据集中无重复数据。

2.2.2 缺失值检测与处理

1、缺失值检测

对数据集的属性特征进行缺失值检测，结果如下表所示：

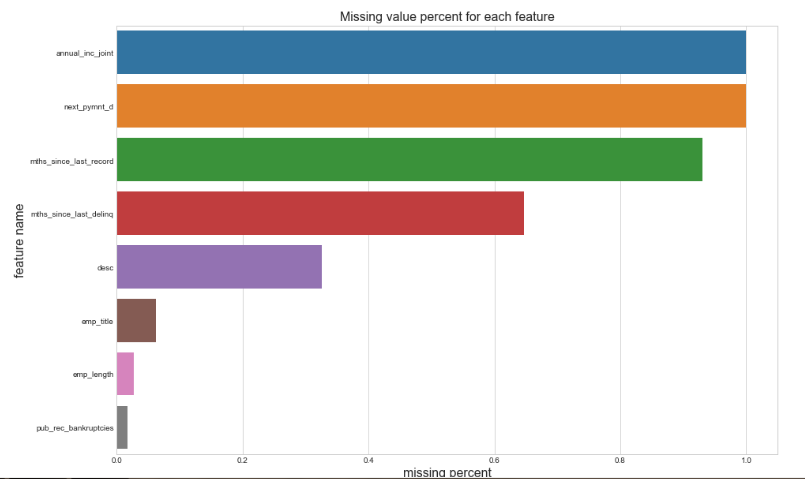
|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 属性 | 缺失数量 | 缺失比例 |
| 1 | annual\_inc\_joint | 39786 | 1 |
| 2 | next\_pymnt\_d | 39785 | 0.999975 |
| 3 | mths\_since\_last\_record | 36994 | 0.929825 |
| 4 | mths\_since\_last\_delinq | 25728 | 0.64666 |
| 5 | desc | 12967 | 0.325919 |
| 6 | emp\_title | 2467 | 0.062007 |
| 7 | emp\_length | 1078 | 0.027095 |
| 8 | pub\_rec\_bankruptcies | 697 | 0.017519 |
| 9 | last\_pymnt\_d | 71 | 0.001785 |
| 10 | collections\_12\_mths\_ex\_med | 57 | 0.001433 |
| 11 | chargeoff\_within\_12\_mths | 56 | 0.001408 |
| 12 | revol\_util | 50 | 0.001257 |
| 13 | tax\_liens | 40 | 0.001005 |
| 14 | title | 11 | 0.000276 |
| 15 | last\_credit\_pull\_d | 2 | 0.00005 |
| 16 | delinq\_amnt | 1 | 0.000025 |
| 17 | acc\_now\_delinq | 1 | 0.000025 |
| 18 | last\_pymnt\_amnt | 1 | 0.000025 |
| 19 | application\_type | 1 | 0.000025 |

由上表可知，

（i）在数据集的55个人特征属性中，共有19个属性具有缺失值，属性中包含缺失值的占比为34.5455%。

(ii)属性缺失比例分布不均，有的甚至全部缺失，有的缺失仅有1个数据；

同时通过数据缺失比例（缺失值比例大于0.001785的特征）可视化图，如下：

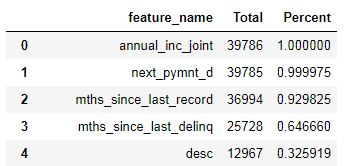


因此下面将针对的不同缺失比例对数据进行处理。

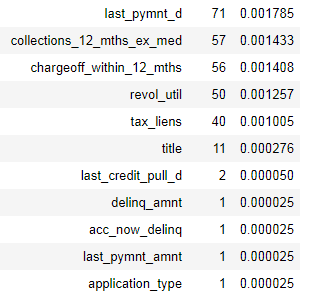
2、缺失值处理

1. 删除缺失比例较大的属性列

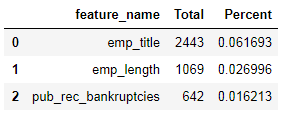
由数据缺失值表，可以看出有的属性，例如：annual\_inc\_joint全部缺失，这样的属性只能考虑删除，保留这类缺失比例较大的属性不仅增加了之后建模的难度，也没有保留过多有用的属性信息，反而显得冗杂；

因此，最终考虑删除缺失比例大于0.15的特征，这五个特征为:

1. 删除缺失比例较少的属性数据所在的行

由数据缺失值表，可以看出有的属性，例如：application\_type仅缺失1个数据，因此对此类属性，考虑将缺失值所在的行删除，最终考虑删除缺失比例小于0.017519的特征的缺失值所在行数据，这11个特征为:（列名同上）

1. 填补缺失比例居中的属性数据值

经过两次属性数据的删除，现在只剩下三个具有缺失值的属性，如下：

下面结合属性的数据类型，分别对三个属性采用了两类填补方法：

（i）用均值填补

pub\_rec\_bankruptcies数据类型为浮点型，因此可以用均值进行642个缺失值填补；

（ii）用众数填补

emp\_title、emp\_length数据类型为object，非数值型，这两个属性分别代表了借贷人的工作和工作年限，因此此处考虑用众数填补缺失值。

至此，数据预处理过程提升了数据的质量和表达，我们的数据集为：loan\_data.csv有39599行用户记录，其中有51个特征属性。

2.3 特征工程

特征工程是将原始数据转换为更能代表预测模型的潜在问题的特征的过程，可以通过挑选最相关的特征，提取特征以及创造特征来实现。在论文采用的数据集中可能面对的问题有：特征之间有相关性，特征和标签无关，特征太多或太小，或者干脆就无法表现出应有的数据现象或无法展示数据的真实面貌。特征工程的目的：

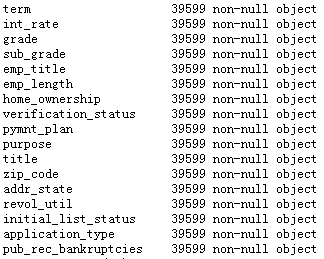
1) 降低计算成本，

2) 提升模型上限

（1）特征衍生

Lending Club平台中，"installment"代表贷款每月分期的金额，我们将'annual\_inc'除以12个月获得贷款申请人的月收入金额，然后再把"installment"（月负债）与（'annual\_inc'/12）（月收入）相除生成新的特征'installment\_feat'，新特征'installment\_feat'代表客户每月还款支出占月收入的比，'installment\_feat'的值越大，意味着贷款人的偿债压力越大，违约的可能性越大。

（2）特征抽象

对于数据集中存在的非数值型数据，采用编码的方式进行处理，包含的属性如下：

考虑采用独热编码，会导致数据过于高维，并且会带来很大的稀疏性，因此最后对上述属性进行直接编码。

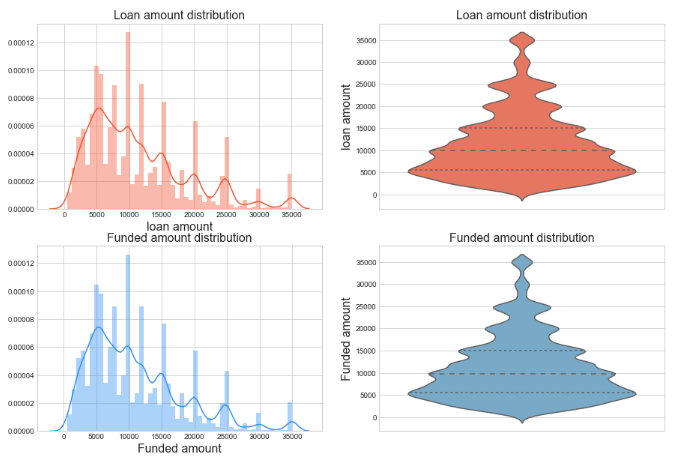
至此，采用了特征工程深入挖掘数据内部的信息，两者结合起来对数据进行操作，可以让所得数据适应模型，匹配模型的需求，更好地服务于之后采用的算法模型。最终得到的loandata.csv有39599行用户记录，其中有52个特征属性。将用于下面不同模型的训练和测试。

3 数据可视化

3.1 借贷信息特征分析

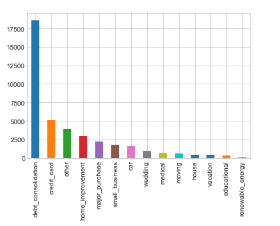
3.1.1 贷款特征

特征loan\_amnt和funded\_amnt分别代表每笔贷款申请的贷款金额和实际贷款的金额，因此这两个特征的数值应当是一致的。下面单独提取出这两个特征，分别画出每个特征的数据分布图，如下：



由图像，可以看到特征loan\_amnt和funded\_amnt的分布确实是一致的，说明这两个数据分布一致，同时我们注意到大部分贷款额度都在20000以下。两个特征的分布呈现右偏趋势。

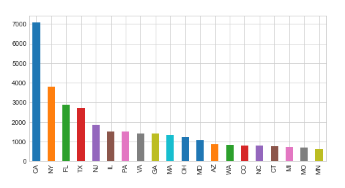
3.1.2 贷款目的分布

根据特征purpose（贷款目的）画出图像如下：

由图像可知，借贷者因为debt\_consolidation借款的数量最大，其次是credit\_card，可以看出贷款主要是为了债务整合和信用卡偿还（债务整合就是借信用卡还其他信用卡）；

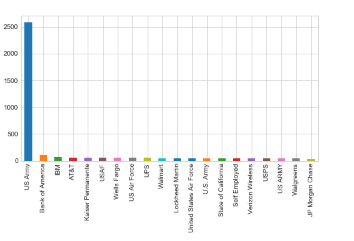
3.2 客户画像

3.2.1 客户的地域分布

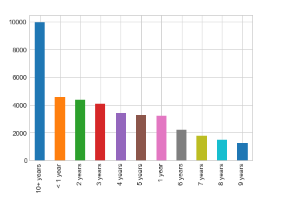
根据特征addr\_state画出借贷用户数所在的地域分布图，如下：

由图像可以看出用户地域分布较广，不过大部分地区借贷人数差别不是很大；

3.2.2 职业分布前20

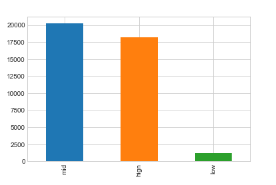
根据特征emp\_title画出借贷用户的前20的职业分布，如下：  
由图像可以看出，各行各业的借贷人都有，居然是US Army（美国军队），其他行业的借贷人员数量差别不太大；

3.2.3 职业年限分布

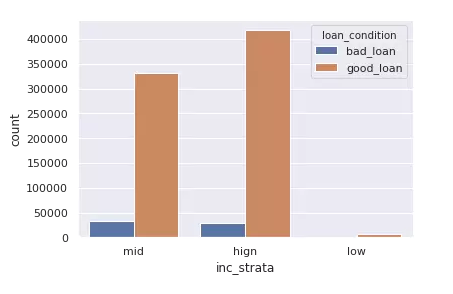
根据特征emp\_length画出借贷用户的职业年限的分布，如下：

由图像可以看出，贷款人数最多的是工作年限越多的，因此我们可以推测工作年限越长越容易贷款；

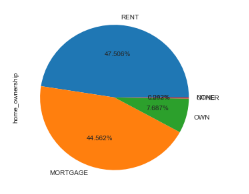
3.2.4 用户年收入（美元）分布

根据特征emp\_length画出借贷用户的年收入分布，这里将年收入大致分为三个区间：20000以下的视为低年收入，20000-60000视为中等，高于60000的就是高收入人群，结果如下：

由图像可以看出，大部分参与借贷客户年收入都在20000以上，下面查看一下贷款质量与年收入之间的关系图：

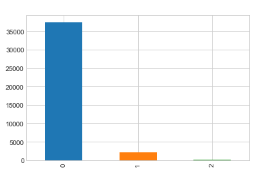
由图像可以看出，中等人群坏账数量最多，因此此类人群是借贷公司在收回贷款时需要特别关注的群体。

3.2.5 用户住房情况

根据特征home\_ownership画出借贷用户的住房情况，如下：

可以看出借贷用户中比例较大的用户的住房主要是依靠：RENT和MORTGAGE两种状态。

3.2.6 用户损坏公物情况

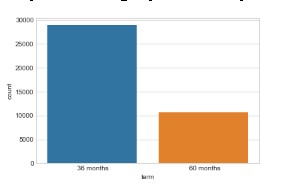
根据特征home\_ownership画出借贷用户的贬损公共记录的次数，如下：

由图像可以看出，借贷用户的贬损公共记录的次数愈多的越不容易申请贷款，也因此包含此类的贷款人数很少，看来有不良记录的人很难申请贷款。

3.3 借贷业务分析

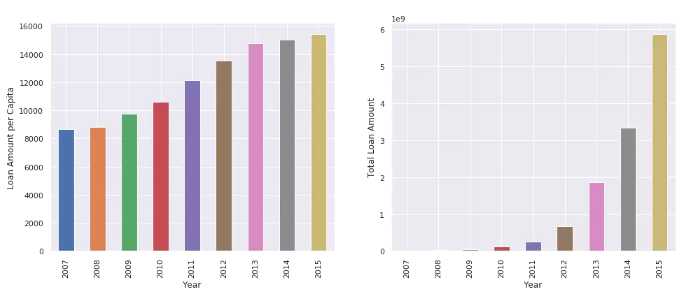
3.1和3.2分别从借贷信息和客户两个角度去挖掘信息，可以看出两者借贷行为的产生跟客户的自身信息存在很大的关系，同时也可以看出借贷金额的成功偿还也与客户的行为存在很大关系，因此结合两者进一步挖掘信息是值得的。

3.3.1 贷款期限分布

根据特征term（借贷年限）画出借贷用户的贷款期限分布，如下：

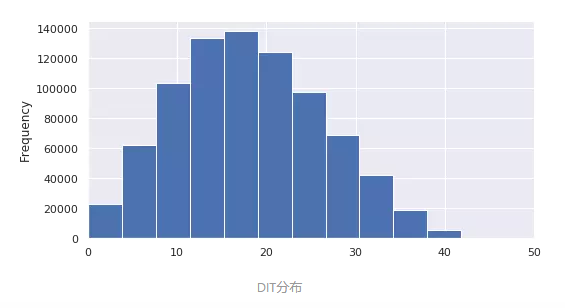
由图像可以看出，大部分借贷用户会选择36months的借贷期限，这也可以看出Lend Club平台以短期贷款为主，但长期贷款比例也不低。

3.3.2 每年人均贷款总额，年均贷款总金额

根据特征issue\_year（还款年限）和loan\_amnt（借贷金额）画出借贷用户的贷款分布，如下：

可以看出在2012年到2015年之间，借贷平台借出的金额越来越多，说明Lend Club平台在2012-2015飞速发展，参与借贷的资金越来越多；

3.3.3 每月还款占月收入的比例

根据特征dit画出借贷用户的每月还款占月收入的比例，如下:

由上图可以得出：

（1）大部分的贷款客户的dit在35%以下，说明还款压力不是很大  
（2）一小部分客户dit达到45%，存在风险  
（3）在右侧看不见的地方还存在极小一部分，，基本属于风险很大的贷款

根据上述数据可视化的结果，一方面从平台自身分析：可以看出借贷平台近年在不断的发展，尤其是可以用于借贷的资金也在逐年增长，说明这个行业具有不错的发展前景；另一方面从用户身上分析，可以看出用户的职业和用户的各种消费和公共行为都会影响其借贷需求；最后结合两方面的因素，合情地分析用户的借贷行为是制定合理的借款条款和提升还款的质量的关键。为了更好的挖掘数据中隐藏的信息，下面采用三种不同的分类算法进行建模。

4 模型的建立与测试

4.1 模型的数据集划分

在数据探索性分析中，提到过运用于模型的数据呈现出不平衡的表现，即计算得到正负标签之比：

**(Y=0): (Y =1)= 6.09913947651488:1**

因此论文中划分的测试集和训练集的时候会以预测标签y为策略进行分层抽样，否则会仅有少数的正样本大部分出现在训练集或者测试集，其中最后得到的划分比例为：

**训练集数据量 ：测试集数据量 = 7 ：3**

4.2 模型的评价指标AUC

论文采用的数据集，具有数据不平衡的现象，因此考虑模型评价指标为：AUC。AUC最直观的理解为：随机挑选一个正样本以及一个负样本，算法根据计算得到的score值将这个正样本排在负样本前面的概率，即是AUC值。score值就是预测为正的概率的值，排在前面表示的是正样本的预测为正的概率值大于负样本的预测为正的概率值。本文中采用了直观地绘制ROC曲线，计算ROC曲线下面的面积估计AUC的值，由此展示论文中模型在训练和测试上的表现好坏。

ROC曲线：接收者操作特征(receiveroperating characteristic),roc曲线上每个点反映着对同一信号刺激的感受性。横轴：假正类率(false postive rate FPR)特异度，划分实例中所有负例占所有负例的比例；(1-Specificity)；纵轴：真正类率(true postive rate TPR)灵敏度，Sensitivity(正类覆盖率)。

针对一个二分类问题，将实例分成正类(postive)或者负类(negative)。但是实际中分类时，会出现四种情况.

(1)若一个实例是正类并且被预测为正类，即为真正类(True Postive TP)

(2)若一个实例是正类，但是被预测成为负类，即为假负类(False Negative FN)

(3)若一个实例是负类，但是被预测成为正类，即为假正类(False Postive FP)

(4)若一个实例是负类，但是被预测成为负类，即为真负类(True Negative TN)

由此可以得到一个二分类问题的混淆矩阵：

|  |  |  |  |
| --- | --- | --- | --- |
| 样本 | | 预测值 | |
| 1 | 0 |
| 实际值 | 1 | TP（真正） | FN（假负） |
| 0 | FP（假正） | TN（真负） |

可以知道二分类ROC曲线横纵轴值，计算方式为：

横轴：（FPR）= FP/(FP+TN)

纵轴：（TPR）= TP/(TP+FN)

我们使用了jupyter notebook中的plotROC绘制ROC曲线，roc\_auc\_score估计AUC值，同时计算出混淆矩阵，下面开始对数据进行建模。

4.3 随机森林模型原理

集成学习（ensemble learning）是时下非常流行的机器学习算法，它本身不是一个单独的机器学习算法，而是通过在数据上构建多个模型，集成所有模型的建模结果。本次主要采用了随机森林（装袋法典型）集成算法，其形象理解表示如下图所示：

其中：

（1）装袋法的核心思想是构建多个相互独立的评估器，然后对其预测进行平均或多数表决原则来决定集成评估器的结果。装袋法的代表模型就是随机森林。

（2）提升法中，基评估器是相关的，是按顺序一一构建的。其核心思想是结合弱评估器的力量一次次对难以评估的样本进行预测，从而构成一个强评估器。提升法的代表模型有Adaboost、GBDT、Xgboost。

集成算法的核心是要解决以下问题：

（1）如何保证集成后的分类效果好于单个基分类器的效果？

Sklearn库中随机森林参数的调整： n\_estimators：这是森林中树木的数量，即基评估器的数量。这个参数对随机森林模型的精确性影响是单调的，n\_estimators越大，模型的效果往往越好。对其画学习曲线来确定的最优取值，

4.4 Xgboost算法的原理

XGBoost全称是Extreme Gradient Boosting，译为极限梯度提升算法。XGBoost本身的核心是基于梯度提升树实现的集成算法，整体来说可以有三个核心部分：集成算法本身，用于集成的弱评估器，以及应用中的其他过程。它能够比其他使用梯度提升的集成算法更加快速，并且已经被认为是在分类和回归上都拥有超高性能的先进评估器。

XGBoost算法一般采用正则化+泰勒展开学习得到目标函数最优化，目标函数是自身的损失函数和正则化惩罚项相加而成的，由此算法能够防止过拟合，高维稀疏等影响模型效果的消极因素；

其中，目标函数一般形式为：

Object=

第一项为损失函数，第二项为正则惩罚项；

求解方法采用加性训练，大致如下：



Object=

再结合泰勒展开，最后我们的目标函数可以转化为：

Object=

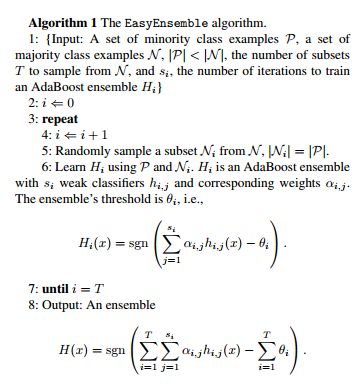
其中：

XGBoost背后也是CART树，这意味着XGBoost中所有的树都是二叉的。其思想是不断地添加树，不断地进行特征分裂来生长一棵树，每次添加一个树，其实是学习一个新函数，去拟合上次预测的残差。当我们训练完成得到k棵树，我们要预测一个样本的分数，其实就是根据这个样本的特征，在每棵树中会落到对应的一个叶子节点，每个叶子节点就对应一个分数，最后只需要将每棵树对应的分数加起来就是该样本的预测值。

4.5 EasyEnsemble算法原理与应用

easy ensemble算法每次从多数类中抽样出和少数类数目差不多的样本，然后和少数类样本组合作为训练集。在这个训练集上学习一个adaboost分类器。  
 最后预测的时候，是使用之前学习到的所有adaboost中的弱分类器（就是每颗决策树）的预测结果向量（每个树给的结果组成一个向量）和对应的权重向量做内积，然后减去阈值，根据差的符号确定样本的类别。

下面展示算法的步骤：



4.6 三种模型结果对比分析

上面针对高维稀疏和不平衡数据问题，我们采用了Xgboost算法，easy ensemble算法以及随机森林算法进行建模和测试，展示测试效果，结果如下表：

|  |  |  |
| --- | --- | --- |
| 算法 | 运行时间 | AUC值 |
| 随机森林算法 | 2.6871535778045654s | 0.9999484489645343 |
| Xgboost算法 | 4.758271932601929s | 0.9997858923564462 |
| easy ensemble算法 | 370.70320296287537s | 0.999331589571276 |

从时间成本和运行AUC值分析：

一、XGBoost 运行效率的提升来自两方面，分别是算法理论和工程实现。算法理论上主要是 Column Subsampling， Histogram-based Algorithm(特征分割点-> 分割区间) 以及 Sparsity-aware Split Finding，减少了计算量。更主要的是工程上的实现带来了效率提升, 底层由 C 和 C++ 实现， Cache-aware block structure + Out-of-core tree learning 且支 持并行 (如寻找最优分割时)。

二、 EasyEnsamble 慢的原因主要是工程实现。因为所用库 imblearn 由 Python 实现，虽然对并行支持很好，也是无法弥补与 C++ 效率上的差距。

三、三种算法在测试集上的表现都非常好，都接近于1，因此三种模型都适用于该数据集所代表的一类实际问题的建模分析。

5 结论

从金融借贷市场的发展现状来看，更好的预测借贷信用风险是一个极其重要的问题，关乎金融借贷系统运行的稳定。论文针对客户借贷状态类型的预测，首先对于收集到的实际数据，采用了数据预处理和特征工程，进一步提升了训练模型的数据质量，对于提高模型效率的上限有一定帮助；然后分别采用了Xgboost算法、easy ensemble算法和随机森林算法对数据进行建模和训练，最后通过分析总结三种模型的测试结果，认为论文中采用的三种算法训练得到的模型均可以用于实际预测，为借贷平台公司制定更优质的借贷条例和进一步降低借贷者还款风险提供了理论模型的指导意义。