第一届“信贷用户逾期预测”算法大赛

模型说明文档

参赛选手：汤杰强

2018 年 5 月

摘要

随着金融机构资产日益多样化、互联网金融等的兴起， 信贷资产管理更加复杂，使得银行信贷面临着越来越大的风险，传统的客户信用评估体系已难以应付现实需要。以数据为导向构建模型， 因此它能够更好地捕捉到数据之间的相关性，符合大数据时代的需求，适宜经济周期不稳定、金融环境复杂所引起的数据状况，机器学习法通过算法随时模型捕捉不断累加数据的相关性，计算机归纳建模，省略了人工建模耗时费力的过程。

本文遵循银行信贷业务的基本原则， 通过贷款人所提供的资料，对贷款人的基本情况、还款能力、品质动机、健康状况等逐一解析，利用机器学习的方法建模预测用户贷款预期。主要对数据的缺失值的预处理，对数据不平衡的重抽样以及分别使用逻辑回归模型、支持向量机、随机森林、Adaboost 和深度神经网络模型对数据进行建模。

目录

1 模型解释

本文主要介绍对信贷用户逾期进行预测所涉及到的主要步骤和使用到的主要算法。包含了下面几个大的方面：

1. 数据的可视化直观展示，通过观察数据的分布等情况，对数据有一个更加直观的了解；

2. 原始数据缺失值的处理，通过观察数据，我们发现原始数据中存在大量数据缺失的情况，有些特征数据缺失达到 90%以上，对于数据缺失值的处理，我们采用的是大于一定阈值的数据，我们将采取将那一列特征值数据剔除，而对于在数据缺失容忍度范围内的那些数据，我们采用 Missing Not At Random (MNAR)的方法来进行填充缺失数据，之所以使用这些方法进行填充，是因为我们观察到原始数据的缺失是有规律的，而不是随机的；

3. 原始数据样本的不均衡处理，通过统计，我们发现原始数据中有 80%的数据，用户是属于不违约的，而仅有 20%的数据是预期违约数据。那么这就存在一个样本不平衡的问题。对于样本不平衡问题，有很多的解决方案，其中包括 under-sampling 和 over-sampling,由于考虑到我们的数据集比较小，我们选择用 over-samplig 的方法来解决样本不平衡的问题；

4. 在对数据进行标准化之前，我们发现有关贷款金额，放款金额等，这些涉及到金额有关的数据，我们考虑生成它的衍生变量，对这些数据取对数(log)，这样更符合经济意义；

5. 对数据进行标准化处理，因为在模型训练之前，我们对于特征值都要同等对待，所以要通过标准化处理，使不同量级的数据具

有可比较性。文章中提出来四种标准化处理的方法，Z-Score、Min-Max、分位数法以及 RobustScaler 法，通过实验结果比较，RobustScaler 法具有更好的结果；

6. 特征选择，在之前缺失值处理的时候，我们已经将小部分特征剔除掉了。而要进一步进行特征选择，我们主要考虑到了。如果特征太多将会导致最终训练模型的过拟合，所以选择适当的特征进行训练能一定程度上避免过拟合问题。我们这边主要用到了四个算法来进行特征选择。包括随机森林法、L2 正则化、SBS 算法、主成分分析法(PCA);

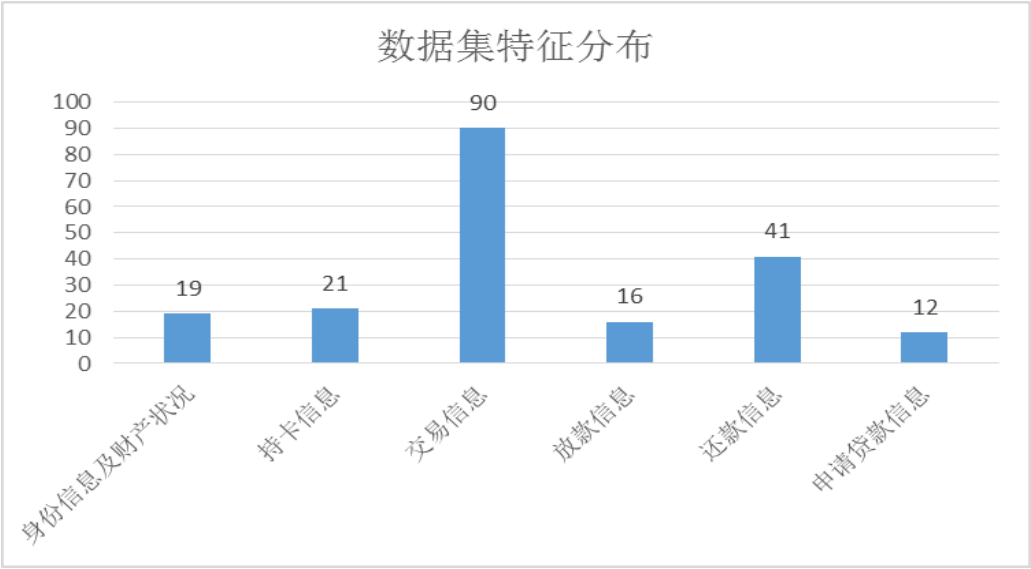
7. 通过数据预处理，特征工程处理之后，我们将进行学习模型的选择。本文中，我们主要用到了一些传统的机器学习学法，比如逻辑回归模型和支持向量机，还用到了集成学习，比如随机森林和 Adaboost,还使用的深度神经网络模型。对于模型训练中存在的过拟合问题，处理减少特征值个数之外，我们还采用了交叉验证，加入惩罚系数以及添加 dropout(这个主要是神经网络避免过拟合的处理方法)

2 建模思路

2.1 数据集描述

要使用机器学习对于特定的任务进行模型训练之前，我们要先对训练任务有一个了解，对于信贷用户逾期的问题以及它的数据集，数据集中一共包含了 199 个特征数据。

数据特征属性	包含特征数目	数据含义
身份信息及财产状况	19	包含了贷款人的基本信息
持卡信息	21	银行卡信息，卡数等
交易信息	90	贷款人银行卡金额流转，以及包括相应的衍生变量，标准差等
放款信息	16	用户近六个月的放款信息
还款信息	41	用户近六个月的还款信息
申请贷款信息	12	用户近六个月的申请贷款的信息



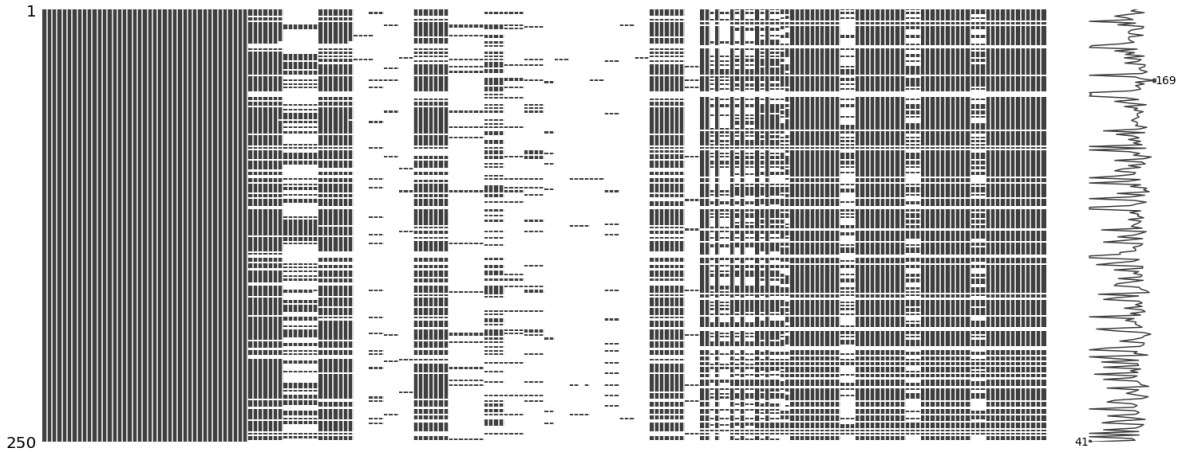
从上图和表中，我们可以发现交易信息和还款信息中特征数目是最多的，但是当我们进一步挖掘其中具体的特征字段时，我们发现很多都是特征字段都是相似或者部分重复的。比如说交易信息中

罗列了很多不同类别的交易，有教育、互联网、汽车类和保险类等。其实某种意义上这些特征字段都具有相同的含义，所以说这些字段是存在多重共线性的可能的。

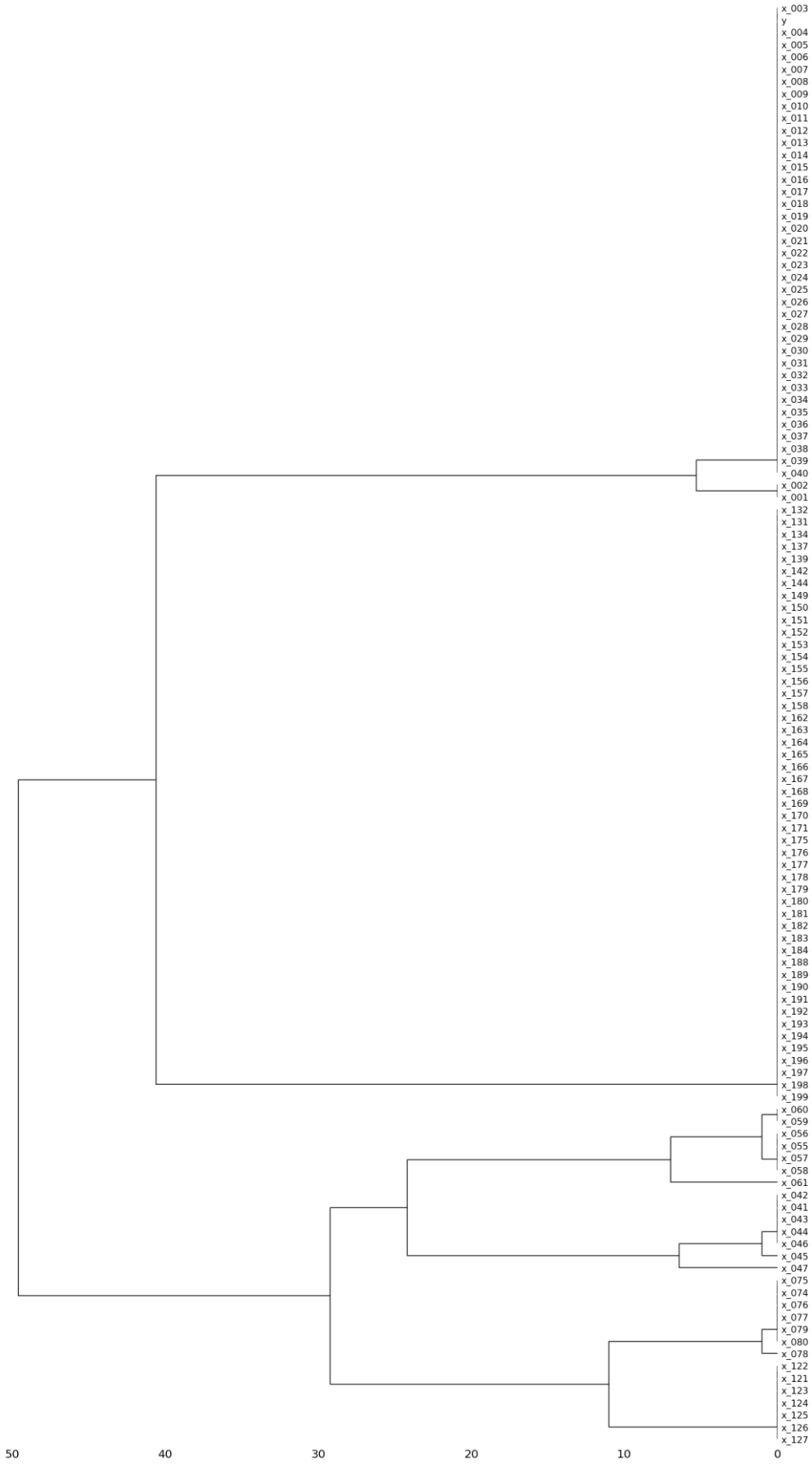
2.2 缺失值处理

通过对原数据的查看，我们发现数据中存在大量的数据缺失情况，这边我们使用 Python 的数据可视化 missingno 包对图像进行分析， missingno 包是用图像的方式让用户能够快速评估数据缺失的情况，而不是在数据表里面单纯看数据。可以根据数据的完整度对数据进行排序或过滤，或者根据热度图或树状图来考虑对数据进行修正。.

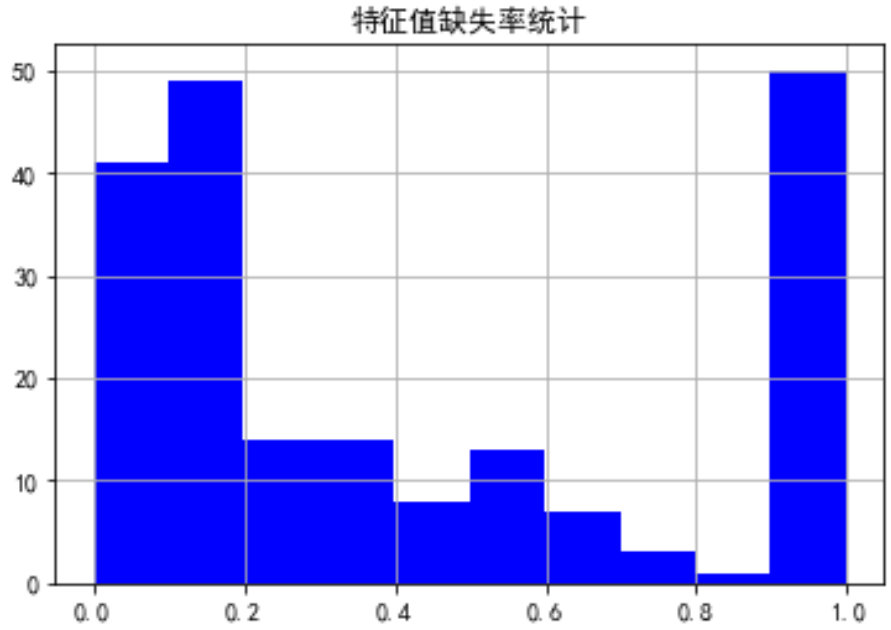
- **数据矩阵**



- **数据树状图**


x_003
y
x_004
x_005
x_006
x_007
x_008
x_009
x_010
x_011
x_012
x_013
x_014
x_015
x_016
x_017
x_018
x_019
x_020
x_021
x_022
x_023
x_024
x_025
x_026
x_027
x_028
x_029
x_030
x_031
x_032
x_033
x_034
x_035
x_036
x_037
x_038
x_039
x_040
x_002
x_001
x_132
x_131
x_134
x_137
x_139
x_142
x_144
x_149
x_150
x_151
x_152
x_153
x_154
x_155
x_156
x_157
x_158
x_162
x_163
x_164
x_165
x_166
x_167
x_168
x_169
x_170
x_171
x_175
x_176
x_177
x_178
x_179
x_180
x_181
x_182
x_183
x_184
x_188
x_189
x_190
x_191
x_192
x_193
x_194
x_195
x_196
x_197
x_198
x_199
x_060
x_059
x_056
x_055
x_057
x_058
x_061
x_042
x_041
x_043
x_044
x_046
x_045
x_047
x_075
x_074
x_076
x_077
x_079
x_080
x_078
x_122
x_121
x_123
x_124
x_125
x_126
x_127
50
40
30
20
10
0

由于模型数据比较大，我们这边取了前 250 个样本数据，上面的数据矩阵分析图，给我们一个直观的展示，空白处表示数据缺失，从上面我们可以看出，有些特征数据缺失率几乎达到 1 左右，矩阵图右边表示每行的缺失值数据个数，最多一行缺失数据达到 169 个，缺失率为 85%。为了进一步观察我们的缺失数据情况，我们对数据进行缺失值统计，如下图:



特征值缺失率在 90%以上的有将近 50 个，这就意味着有将近四分之一的特征值缺失这么严重。一般认为特征值缺失率达到 40%以上的，我们认为该特征值不具有解释性，或者说该缺失数据对模型预测具有较大的影响，所以将那些缺失率大于 40%的特征剔除掉。对于剩下的 117 个特征字段数据进行进一步分析。

对于这些数据，我们将对其进行缺失值的填充。对于缺失值一般有三种情况：

1） 完全随机缺失（Missing Completely at Random，MCAR）。数据的缺失与不完全变量以及完全变量都是无关的。

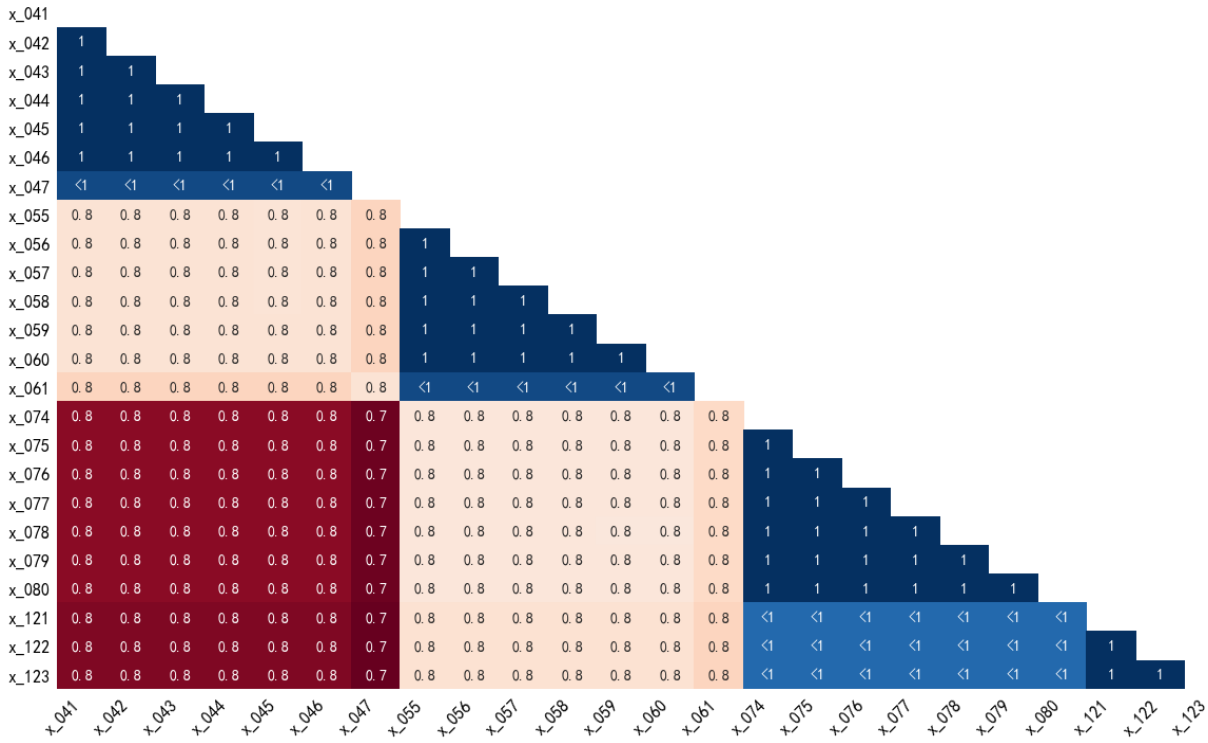
2） 随机缺失（Missing at Random，MAR）。数据的缺失仅仅依赖于完全变量。

3） 非随机、不可忽略缺失（Not Missing at Random, NMAR）。不完全变量中数据的缺失依赖于不完全变量本身，这种缺失是不可忽略的。

通过数据矩阵可视化图，我们发现训练数据缺失情况属于非随机不可忽略缺失的。所以我们将采用与该类缺失情况有关的填充算法，我们基于 KNN（最近邻）方法的填充法是寻找和有缺失值的表达相似的其他数据，通过这些数据的表达值（依照表达相似性加权）来填充缺失值。经过填充之后，我们查看数据的缺失相关系数热力图。

- **缺失相关系数热力图**

缺失相关系数热力图展示了一个变量的存在与否影响另一个变量的存在有多强烈。



通过观察，我们发现经过填充之后的缺失数据，相互之间具有更小的影响，这样可以一定程度地减小数据之间的多重共线性问题。

2.3 衍生变量

通过对数据经济含义的理解，我们对部分特征变量进行处理，进一步生成其衍生变量，我们将对那些包含金额的数据进行取对数处理，这些变量包括以下这些：

变量分类	变量名	变量解释
交易信息	x_045	近 6 个月交易金额
	x_047	近 6 个月交易金额均值/标准差
	x_059	近 6 个月异地交易金额
	x_067	近 6 个月夜间交易金额

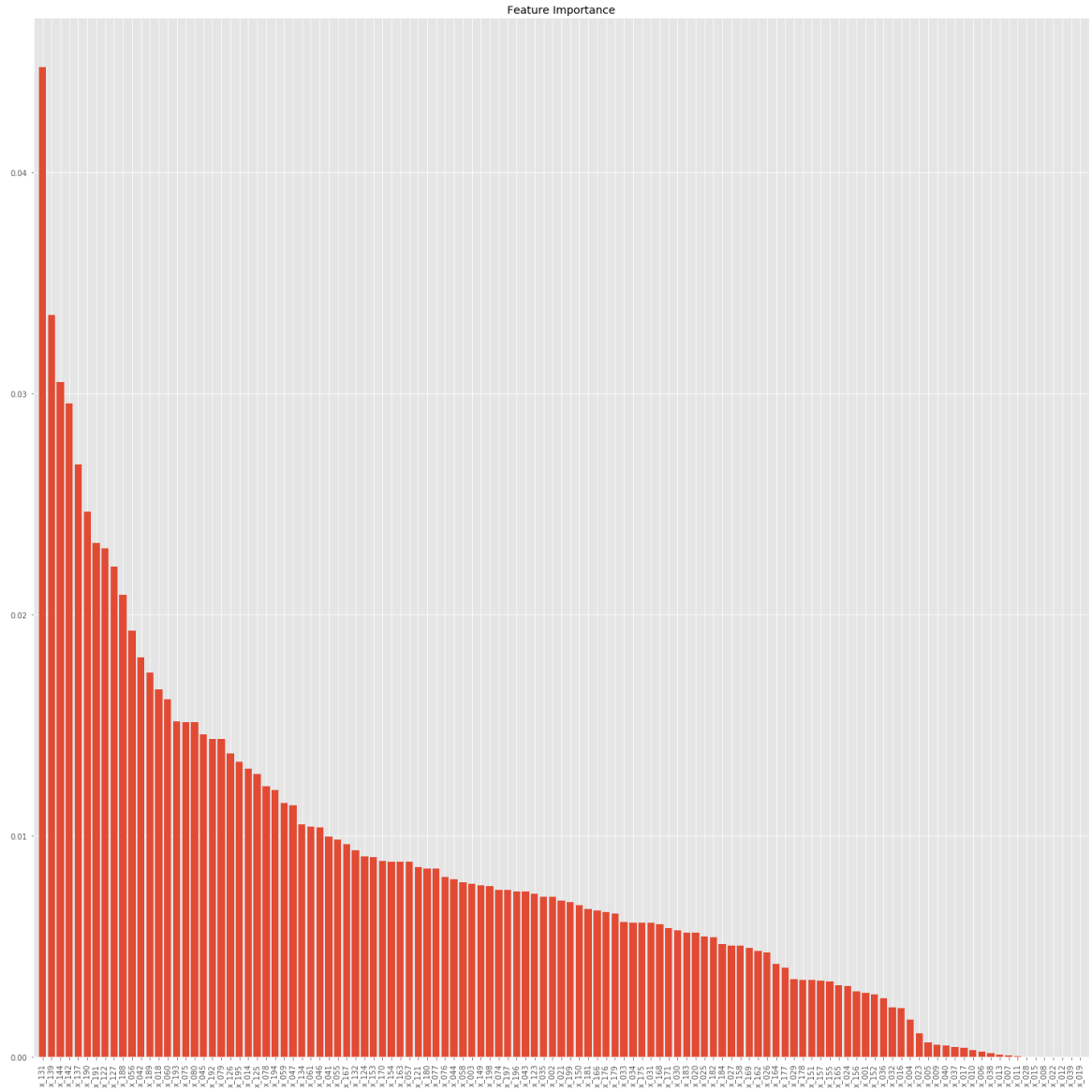
	x_070	近 6 个月公共事业交费金额
	x_073	近 6 个月缴税金额
	x_078	近 6 个月互联网交易金额
	x_085	近 6 个月大额交易金额
	x_130	近 6 个月加油类交易金额
放款信息	x_131	最近一笔放款金额
	x_133	30 天内放款总金额
	x_135	30 天内单笔放款金额最大值
	x_136	30 天内单笔放款金额最小值
	x_138	90 天内放款总金额
	x_140	90 天内单笔放款金额最大值
	x_141	90 天内单笔放款金额最小值
	x_143	180 天内放款总金额
	x_144	180 天内放款总笔数
	x_145	180 天内单笔放款金额最大值
	x_146	180 天内单笔放款金额最小值
还款信息	x_147	最近一笔成功还款金额
	x_148	最近一笔失败还款金额
	x_159	30 天内成功还款金额
	x_160	30 天内单笔还款金额最大值
	x_161	30 天内单笔还款金额最小值
	x_172	90 天内还款成功金额
	x_173	90 天内单笔还款金额最大值
	x_174	90 天内单笔还款金额最小值
	x_185	180 天内成功还款金额
	x_186	180 天内单笔还款金额最大值
	x_187	180 天内单笔还款金额最小值

之所以要对这些数据进行取对数处理，主要的原因包括以下几个方面：

1. 在经济学中，常取自然对数再做回归
2. 缩小数据的绝对数值，方便计算
3. 取对数后，可以将乘法计算转换称加法计算
4. 某些情况下，在数据的整个值域中的在不同区间的差异带来的影响不同
5. 取对数之后不会改变数据的性质和相关关系，但压缩了变量的尺度
6. 所得到的数据易消除异方差问题

2.4 特征筛选

原始数据中一共有 199 个特征字段，通过缺失处理之后，还剩下 117 个特征字段。但是这个级别的特征字段数量，相对于数据样本大小而言，可能导致训练模型的过拟合问题。所以，我们需要对字段进行筛选，取我们认为最有用的小数字段，这边我们取 2*sqrt(117)=22 个字段。我们通过随机森林来筛选特征，该模型可能每次给予特征不同的重要性权重。但是通过多次训练该模型，即每次通过选取一定量的特征与上次特征中的交集进行保留，以此循环一定次数，从而我们最后可以得到一定量对分类任务的影响有重要贡献的特征。通过对我们的数据进行分析，得到特征贡献度图：



通过分析我们筛选出前 22 名的特征，它们分别是

序列	字段代码	贡献度
1	x_131	4.35%
2	x_139	3.15%
3	x_144	3.04%
4	x_142	2.59%
5	x_190	2.55%
6	x_189	2.40%
7	x_137	2.39%
8	x_127	2.32%
9	x_056	2.32%

10	x_191	2.09%
11	x_042	2.07%
12	x_122	2.03%
13	x_194	1.87%
14	x_060	1.76%
15	x_188	1.72%
16	x_075	1.70%
17	x_193	1.66%
18	x_080	1.52%
19	x_079	1.47%
20	x_045	1.44%
21	x_195	1.44%
22	x_126	1.40%

3 模型中的算法

3.1 缺失值处理

由前面分析我们知道原数据缺失属于非随机的，我们分别采用中位数填充、均值填充、KNN 填充、Expectation Maximization 和 Multivariate Imputation by Chained Equations。这边我们对 Expectation Maximization 和 Multivariate Imputation by Chained Equations 进行介绍。

3.1.1 Expectation maximization

该方法比删除个案和单值插补更有吸引力，它一个重要前提：适用于大样本。有效样本的数量足够以保证 ML 估计值是渐近无偏的并服从正态分布。但是这种方法可能会陷入局部极值，收敛速度也不是很快，并且计算很复杂。

EM 算法是一种在不完全数据情况下计算极大似然估计或者后验分布的迭代算法。在每一迭代循环过程中交替执行两个步骤：E 步（Excepctaion step, 期望步），在给定完全数据和前一次迭代所得到的参数估计的情况下计算完全数据对应的对数似然函数的条件期望；M 步（Maximzation step，极大化步），用极大化对数似然函数以确定参数的值，并用于下步的迭代。算法在 E 步和 M 步之间不断迭代直至收敛，即两次迭代之间的参数变化小于一个预先给定的阈值时结束。该方法可能会陷入局部极值，收敛速度也不是很快，并且计算很复杂。

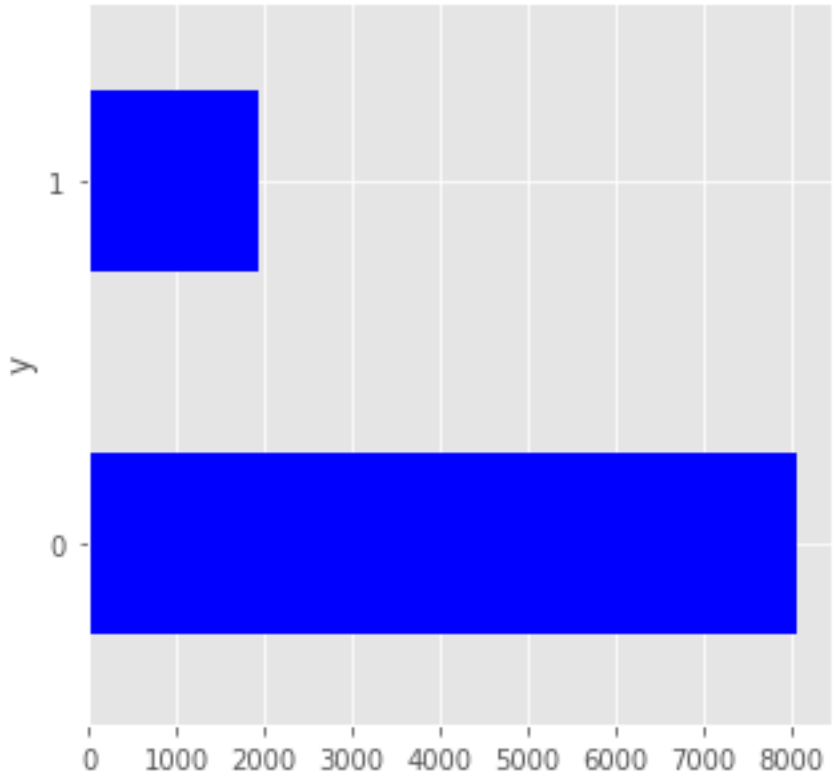
3.1.2 多重填补（Multiple Imputation，MI）

多重插补方法分为三个步骤：①为每个空值产生一套可能的插补值，这些值反映了无响应模型的不确定性；每个值都可以被用来插补数据集中的缺失值，产生若干个完整数据集合。②每个插补数据集合都用针对完整数据集的统计方法进行统计分析。③对来自各个插补数据集的结果，根据评分函数进行选择，产生最终的插补值。

最终选择多重填补法进行填充，更符合数据特征。

3.2 样本不平衡

我们可以查看一下我们的样本分布情况，如下图：



一共是 10000 多个数据，统计结果为违约样本仅仅只有 19%左右，样本明显存在不平衡现象，但是这又是无法避免的问题，现实中确实存在数据就是这样。那么，我们就要对数据进行数据不平衡的处理。一般会有下面几种方法。

3.2.1 过抽样

过抽样（也叫上采样、over-sampling）方法通过增加分类中少数类样本的数量来实现样本均衡，最直接的方法是简单复制少数类样本形成多条记录，这种方法的缺点是如果样本特征少而可能导致过拟合的问题；经过改进的过抽样方法通过在少数类中加入随机噪声、干扰数据或通过一定规则产生新的合成样本，例如 SMOTE 算法。

最后我们选择 SMOTE 算法进行数据合成，具体介绍一下算法

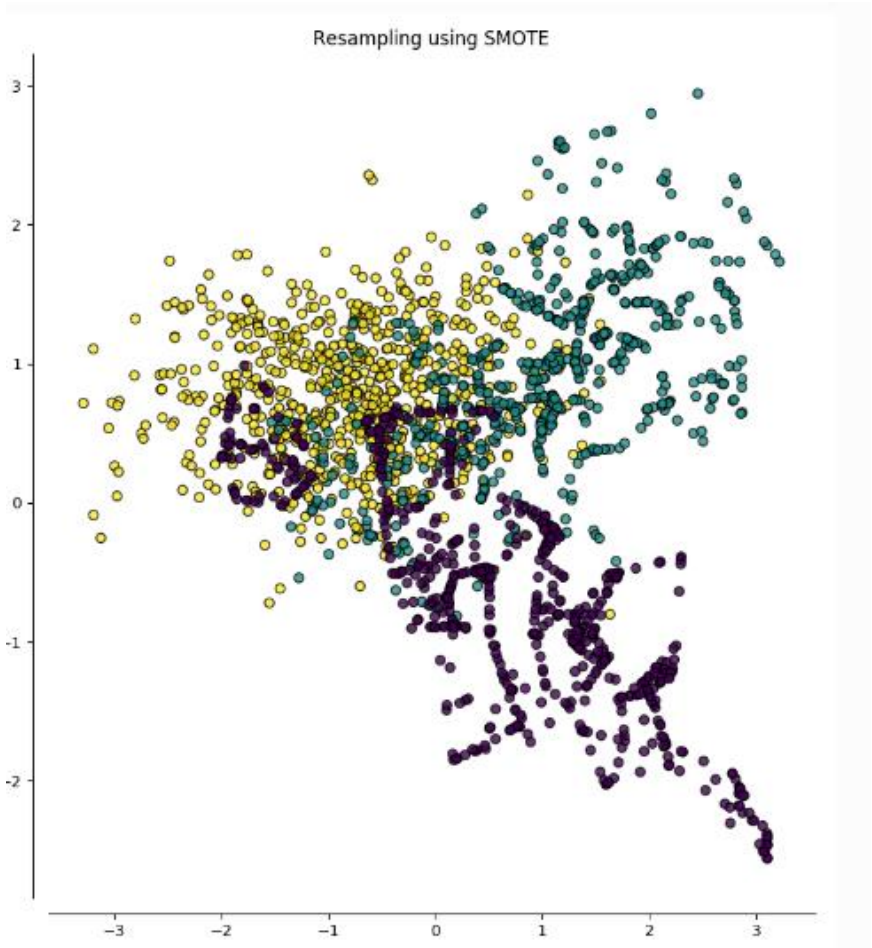
过程以及为什么选用该算法为数据合成。

SMOTE 全称是 Synthetic Minority Oversampling Technique 即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，算法流程如下。

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 Smin 中所有样本的距离，得到其 k 近邻。

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，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

$$x_{new}=x+rand(0,1)*|x-xn|$$



颜色更深的那些数据即为用算法合成的新数据。

3.2.2 欠抽样

欠抽样（也叫下采样、under-sampling）方法通过减少分类中多数类样本的样本数量来实现样本均衡，最直接的方法是随机地去掉一些多数类样本来减小多数类的规模，缺点是会丢失多数类样本中的一些重要信息。

这种方法对于我们的数据集是不适用的，因为我们的数据集相对而言比较小。

3.2.3 正负样本的惩罚权重

通过正负样本的惩罚权重解决样本不均衡的问题的思想是在算

法实现过程中，对于分类中不同样本数量的类别分别赋予不同的权重然后进行计算和建模。

使用这种方法时需要对样本本身做额外处理，只需在算法模型的参数中进行相应设置即可。很多模型和算法中都有基于类别参数的调整设置，以 scikit-learn 中的 SVM 为例，通过在 class_weight : {dict, 'balanced'}中针对不同类别针对不同的权重，来手动指定不同类别的权重。如果使用其默认的方法 balanced，那么 SVM 会将权重设置为与不同类别样本数量呈反比的权重来做自动均衡处理，计算公式为：n_samples / (n_classes * np.bincount(y))。

在我们模型应用中，会综合使用权重均衡和过抽样的方法来解决样本数据不均衡的问题。

3.2.4 组合/集成方法

组合/集成方法指的是在每次生成训练集时使用所有分类中的小样本量，同时从分类中的大样本量中随机抽取数据来与小样本量合并构成训练集，这样反复多次会得到很多训练集和训练模型。

3.3 数据标准化

数据的标准化（normalization）是将数据按比例缩放，使之落入一个小的特定区间。在某些比较和评价的指标处理中经常会用到，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。

在实验中，我们主要用到了三种数据标准化处理算法，Min-Max 标准化、z-score 标准化、分位数标准化。

之所以我们要对数据进行标准化处理，主要有下面几个原因：

1. 提升模型的收敛速度；
2. 提升模型的精度；
3. 深度学习中数据归一化可以防止模型梯度消失。

3.3.1 Min-Max 标准化

$$y_i = \frac{x_i - \min\limits_{1\le j\le n}\{x_j\}}{\max\limits_{1\le j\le n}\{x_j\} - \min\limits_{1\le j\le n}\{x_j\}}$$

min−max 标准化方法是对原始数据的线性变换。设 min 和 max 分别为属性的最小值和最大值，将属性的一个原始值通过 min−max 标准化映射成在区间[new_min，new_max]中的值。

min−max 标准化方法保留了原始数据之间的关系。如果今后输入的数据落在属性的原数据区外，该方法将会面临“越界”错误。

3.3.2 z-score 标准化

$$y_i = \frac{x_i - \bar{x}}{s}，\text{这里}\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i，\ s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

z−score 标准化是基于原始数据的均值（mean）和标准差（standard deviation）进行数据的标准化。将 A 的原始值 x 使用 z−score 标准化到 y。

但是该标准化存在的一个问题为对于极端值没有做处理，可能导致极端值对于最后模型训练的影响很大。

3.3.3 分位数标准化

$$x' = \frac{x - \mu}{\sigma}$$

分位数归一化是使两个分布在统计特性上相同的性质。 要进行分位数 标准化为相同长度的参考分布，先对测试分布进行排序并对参考分布进行排序。 然后，测试分布中的最高条目采用参考分布中的最高条目的值，参考分布中次高的条目等等，直到测试分布是参考分布的扰动。

最后，我们选择用分位数标准化对我们的数据进行标准化，之所以采用这种方法，是因为该方法既考虑了数据压缩到同一比例的可比较性，而且该标准化可以避免异常值的影响。

3.4 特征变量选择

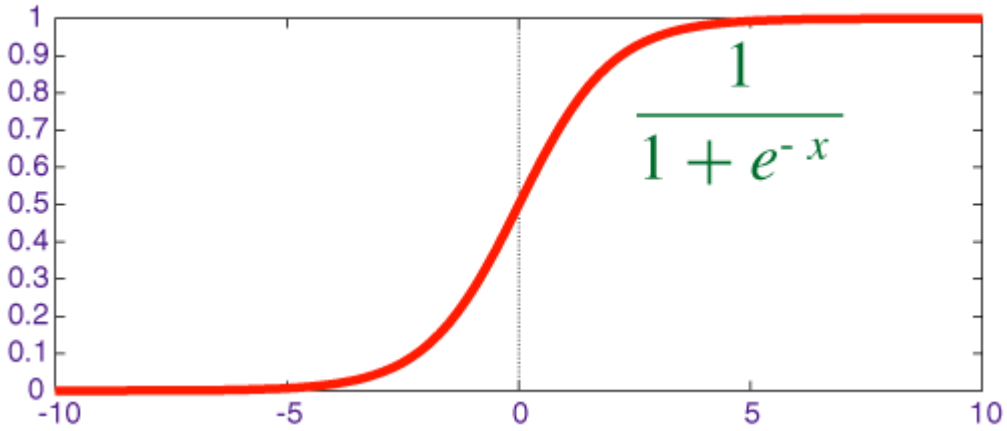
由于特征变量太多容易造成训练模型的过拟合，所以我们要采取一定的方法对特征变量进行选取，我们主要用到了主成分分析（PCA）、随机森林法、L2 正则化。最后，我们选择了随机森林法进行特征值的筛选，详细可参考前文。

3.5 模型训练算法

下面我们将介绍本试验中，用到的几个常用的训练模型算法，分别是 Logistic

3.5.1 Logistic 回归

Logistic 回归主要对分类问题进行回归，考虑到我们的目标是做分类问题，所以我们自然想到用 Logistic 回归来解决这个问题，该回归用 sigmoid 函数作为目标函数，将分类结果映射到 0 和 1 区间，设置阈值能较好的区分开。



我们选用的代价函数为交叉熵代价函数

$$C = -\frac{1}{n}\sum_{x}[y\ln a + (1-y)\ln(1-a)]$$

在用该算法进行训练的时候，我们要注意以下几个方面：

1. 在对测试集数据进行验证的时候，我们取模型预测的概率，而不是模型预测的值。也就是说到时候我们根据预测概率和阈值来确定分类问题。
2. 为防止模型过拟合的问题，我们之前已经对特征项减少来避免这个问题，同时，我们将使用 L2 正则化规格化这个问题。

3.5.2 SVM

SVM 算法最初是为二值分类问题设计的，我们求解的就是一个二分类问题。一般有两种方法来解决这个问题：

（1）直接法，直接在目标函数上进行修改，将多二分类面的参数求解合并到一个最优化问题中，通过求解该最优化问题“一次性”实现多类分类。这种方法看似简单，但其计算复杂度比较高，实现起来比较困难，只适合用于小型问题中；

（2）间接法，主要是通过二分类器来实现。

这边我们使用的是第一种方法，主要考虑到我们的训练数据相对较小。使用直接发能够提升模型自身的准确性。

3.5.3 随机森林

随机森林是一个包含多个决策树的分类器， 并且其输出的类别是由个别树输出的类别的众数而定。随机森林属于集成学习（Ensemble Learning）中的 bagging 算法。在集成学习中，主要分为 bagging 算法和 boosting 算法。

随机森林的一般步骤为：

1. 用 N 来表示训练用例（样本）的个数，M 表示特征数目。
2. 输入特征数目 m，用于确定决策树上一个节点的决策结果；其中 m 应远小于 M。
3. 从 N 个训练用例（样本）中以有放回抽样的方式，取样 N 次，形成一个训练集（即 bootstrap 取样），并用未抽到的用例（样本）作预测，评估其误差。

4. 对于每一个节点，随机选择 m 个特征，决策树上每个节点的决定都是基于这些特征确定的。根据这 m 个特征，计算其最佳的分裂方式。
5. 每棵树都会完整成长而不会剪枝，这有可能在建完一棵正常树状分类器后会被采用）。

那为什么我们选用随机森林来实现我们这个分类问题呢？ 主要是因为随机森林有以下几个有点：

1. 具有极高的准确率
2. 随机性的引入，使得随机森林不容易过拟合
3. 随机性的引入，使得随机森林有很好的抗噪声能力
4. 能处理很高维度的数据，并且不用做特征选择
5. 既能处理离散型数据，也能处理连续型数据，数据集无需规范化
6. 训练速度快，可以得到变量重要性排序
7. 容易实现并行化

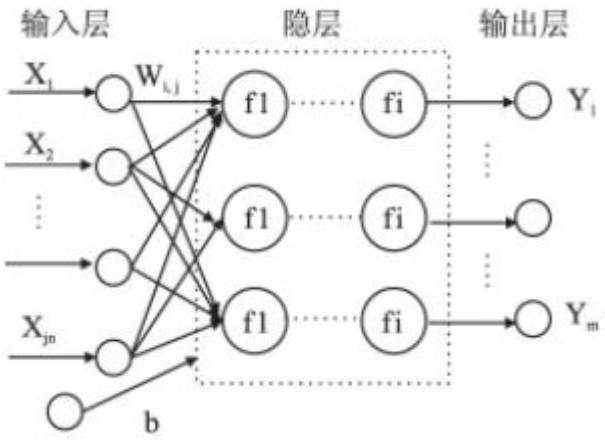
随机森林算法确实是一个很好的学习器，但是同时也存在过拟合的问题，上面已经提及到了随机性能避免过拟合问题，但是除此之外，我们还会做进一步的避免过拟合的处理，我们会对每一个决策树都进行剪枝操作。

3.5.4 深度神经网络

深度神经网络算法已在语音、 图像识别、 广告预测等应用领域取得众多成果． 该算法极大地提升了模式识别算法的识别精度。毫无疑问，巨大的计算量成为深度神经网络研究领域发展的巨大障

碍。由于深度神经网络系统拥有众多的神经元，且训练过程的算法复杂度为 o(n3)。作为深度神经网络算法的重要组成部分，误差反向传播算法(back propagation algorithm，BP 算法) 占据了训练过程中绝大多数的计算时间。所以我们下面主要介绍一下 BP 算法的工作流程。

典型的用于 BP 算法的神经网络(BP 神经网络) 可以分为输入层、 隐藏层和输出层三大部分． 在一个 BP 神经网络中可以拥有多个隐藏层，其网络拓扑结构如图所示：



图中，W 为神经突出全值， b 为偏置或者可称之为阈值，而 f( ) 则为神经网络的激活函数． BP 算法需要通过大量的迭代训练，利用梯度下降的方法来训练得到相对较优的 W 及 B 参数． W 和 B 通过每次的迭代得到的误差， 并利用 delta 法则不断进行修正，最终得到一个较为稳定的解． 以 W 为例，其迭代过程可表示为:

$$W_{i+1} = W_i - \eta \nabla E(W_i)$$

公式(1) 中 W 代表每一层神经网络的权重， i 则表示当前迭代的次数， η 为学习率， 其决定了训练过程的收敛速度， 而 E

则为输出结果和预期值之间的误差.

在 DNN 的训练过程中， 一定数量的样本个数用来组成一个 Minibatch，设每个样本的维数为 D， 样本个数为 N， 则每个 Minibatch 为 D × N 的矩阵. 整体的训练过程则将经历前向计算，后向计算和权重更新三个部分. 设神经网络的层数为 K，在前向计算过程中， 通过输入的 Minibatch 计算得到训练后的输出层 Z，其过程可表示为:

$$\begin{cases} y_{k,j}^{(1)} = F_1\left(\sum_{i=1}^{D} x_{k,i} \times w_{j,i}^{(1)} + b_j^{(1)}\right), k \in [1,N] \\ y_{k,j}^{(\lambda)} = F_1\left(\sum_{i=1}^{NH} y_{k,i}^{(\lambda-1)} \times w_{j,i}^{(\lambda)} + b_j^{(\lambda)}\right), \\ \qquad k \in [1,N], \lambda \in [2,\rho-1] \\ z_{k,j} = S\left(\sum_{i=1}^{NH} y_{k,i}^{(\rho-1)} \times w_{j,i}^{(\rho)} + b_j^{(\rho)}\right), k \in [1,N] \end{cases}$$

在上式子中 x 代表输入的 Minibatch 数据，即输入层数据，y(λ)代表第 λ 层的隐层结果， w （ λ）和 b （ λ）则分别代表第 λ 层网络的权重和偏置， 而 z 则代表前向计算的输出值. 除此之外，F1 （ ）和 S( ) 均为前向的激活函数， 其分别对应前向计算过程中两种不同的操作.

后项计算和权重更新联合实现 delta 法则，在后向计算中通过输出层 Z 与督导层 T 的比较计算出该次迭代的误差量，算法根据 delta 规则对 W 阵值进行更新， 该过程近似于前向和后向计算，也需要经过量乘向量的计算获得更新所需的偏差 Δw 值. 所以，通过观察不难发现， 对于以 Minibatch 作为计算单元的 BP 训练过程可以归结为 3ρ 次的矩阵乘加及其他激活及修正函数的计算过

程。

那么对于这么强大深度神经网络算法，如何避免它的过拟合问题，目前主流的处理办法有数据增强、提前终止和 dropout 处理。

- **数据增强**

让模型泛化的能力更好的最好办法就是使用更多的训练数据进行训练, 但是在实践中, 我们拥有的数据是有限的, 解决这一问题可以人为的创造一些假数据添加到训练集中.

- **提前终止**

当随着模型的能力提升, 训练集的误差会先减小再增大, 这样可以提前终止算法减缓过拟合现象.

- **Dropout**

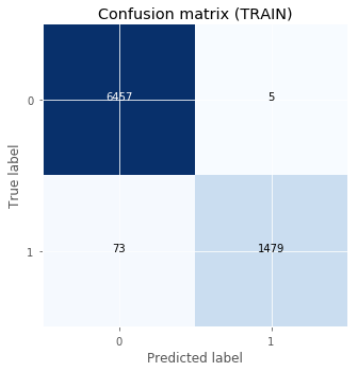
Dropout 提供了一种廉价的 Bagging 集成近似, 能够训练和评估指数级数量的神经网络。dropout 可以随机的让一部分神经元失活, 这样仿佛是 bagging 的采样过程, 因此可以看做是 bagging 的廉价的实现. 但是它们训练不太一样, 因为 bagging, 所有的模型都是独立的, 而 dropout 下所有模型的参数是共享的。

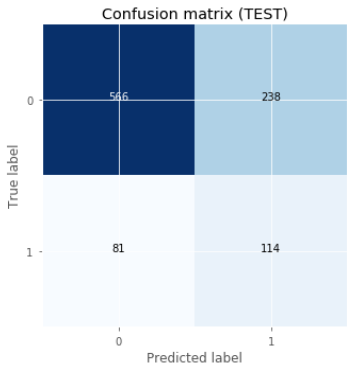
4 算法优化

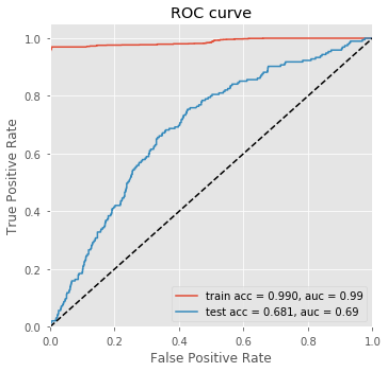
4.1 算法选择

下面列举了模型结果展示图

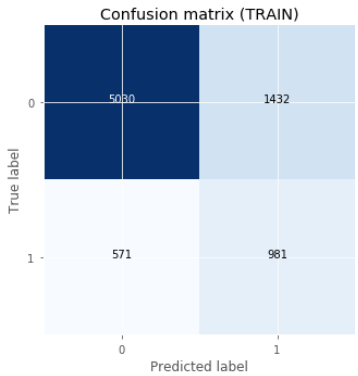
4.1.1 logistics 回归

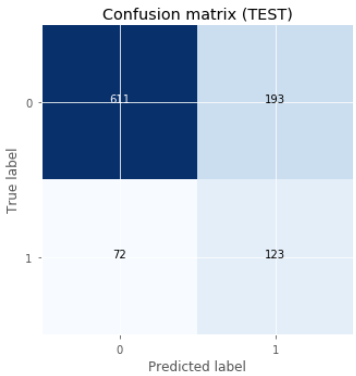


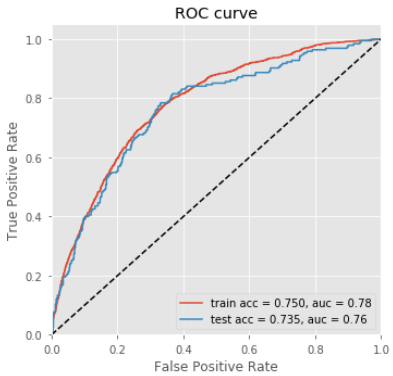




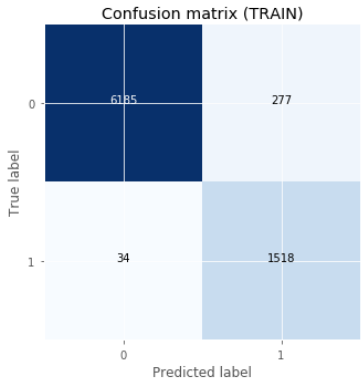
4.1.2 SVM

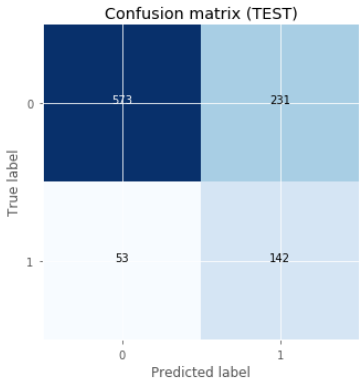


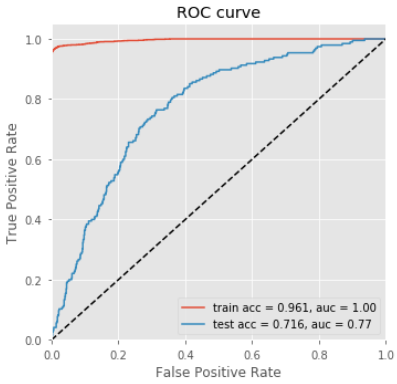




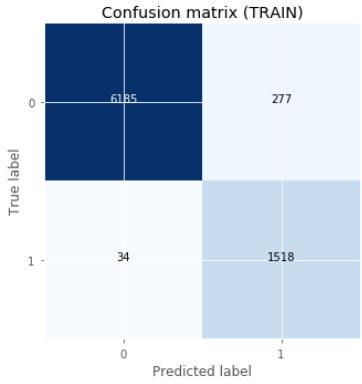
4.1.3 随机森林

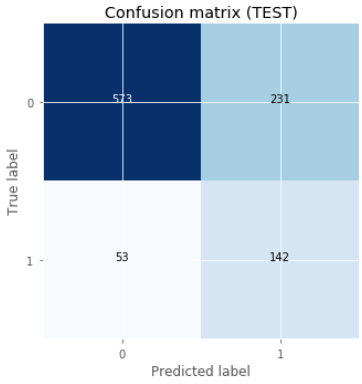


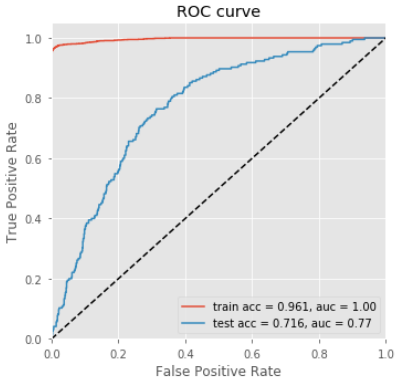




4.1.4 深度学习







4.2 过拟合处理

下面是我在实验过程中使用到的防止过拟合所采用的方法：

算法	过拟合方法
Logigist 回归	减少特征数
	增加数据
	L1, L2 正则化
SVM	减少特征数
	增加数据

	L1, L2 正则化
随机森林	减少特征数
	增加数据
	L1, L2 正则化
	bagging
	剪枝
	随机性
深度神经网络	减少特征数
	增加数据
	L1, L2 正则化
	数据增强
	提前终止
	dropout
	辅助分类节点
	batch normalization

那么，现在我们已经有了避免模型过拟合的方法，为了进一步来验证模型是否过拟合，我们使用交叉验证来验证这一点。

交叉验证（Cross validation），交叉验证用于防止模型过于复杂而引起的过拟合. 有时亦称循环估计， 是一种统计学上将数据样本切割成较小子集的实用方法。于是可以先在一个子集上做分析， 而其它子集则用来做后续对此分析的确认及验证。 一开始的子集被称为训练集。而其它的子集则被称为验证集或测试集。交叉验证是一种评估统计分析、机器学习算法对独立于训练数据的数据集的泛化能力（generalize）

4.3 模型不足点

1、对于 logistics 回归容易欠拟合，一般准确度不太高，所以导致该模型性能不是很好。

2、对于 SVM 算法，对参数和核函数的选择比较敏感，所以不具有泛化性能。对于新样本通常结果会比价差。

3、对于随机森林，对 outlier 比较敏感，就算在一开始我们已经对数据进行了标准化，但是还是不能避免模型对 outlier 预测的准确性。

4、对于深度神经网络，虽然其模型预测结果相对较好。但是模型不具备较强的可解释性，并且这一算一般要求大数据、大计算量。

5 大数据风控

5.1 大数据风控的优势

由于对统计数据的定义越来越明确，使用大数据可以增强风险管理（例如应用程序和行为记分卡）的分析和模型质量。因此，通过加速信息提供，可以更快地制定和解释管理激励措施。考虑到巨大数据量的情景模拟可以有效实现风险集中和对新市场发展的快速反应。大数据还可以在模式识别的帮助下用于欺诈检测，以便通过比较内部和外部数据（例如洗钱或信用卡欺诈）更精确，更快速地识别欺诈并减少手动操作。

使用大数据分析进行金融风险管理的三大优势是速度，效率和更高的可靠性。而在决策过程中加速工作的能力使组织不仅具有竞争优势。使用大量数据更快地工作意味着：

1. 更准确地预测信用风险，更好的业务决策。
2. 根据更长的历史进行贷款和投资决策。
3. 实时获取数据以减少欺诈和违规造成的损失。

5.2 设计产品

1. 它可以改善分析和风险管理模型的质量由于信息加速，可以在几分钟内确定信誉或欺诈风险。
2. 大数据可以提供指示潜在欺诈的模式（例如洗钱，信用卡欺诈等）
3. 大数据可以提供对健康的衍生品风险类型的历史和分析，然后可以使用这些信息来确定潜在投资中类似风险暴露所带来的投资风险。将这些数据放在一个地方，可以减少投资顾问在向客户提供建议之前必须花费的时间。投资组合管理变得更加健全，可以基于客户愿意承担的风险敞口数量。

6 比赛意见与感想

6.1 对大赛建议

对于大赛，我个人有以下几点建议:

1. 建议可以提前说明一下比赛的评选标准及决赛的任务。

2. 建议组委会建立一个官方微信群或者 QQ 群，及时通知比赛

事项。

3. 建议组委会公开参赛队员的得分情况。

6.2 收获

初始阶段建模时提取了很多特征，然后使用一些机器学习算法去预测，但是效果没有达到期望值。随后结合实际问题认真思考，发现其实不一定要使用各种特征，而且很多随机因素对各个特征的影响真的蛮大的，仅使用一些简单的想法也能达到比较好的效果。最后建模完成后，我的感悟是模型真的不是越复杂越好，也不一定要用各种现成的模型，结合实际问题背景去分析可能会比一直纠结各种特征以及模型参数获得更大的收益。

参考文献

[1] Lang, W. (2009). Consumer Credit Risk Modeling and the Financial Crisis. International Atlantic Economic Conference.

[2] Marlin, B. M. (2008). Missing data problems in machine learning. University of Toronto.

[3] Kitchin, R. (2015). The opportunities, challenges and risks of big data for official statistics. Statistical Journal of the Iaos, 31(3), 471-481.

[4] 孙存一, & 王彩霞. (2015). 机器学习法在信贷风险预测识别中的应用. 中国物价(12), 45-47.

[5] Lacković, I. D., Kovšca, V., & Vincek, Z. L. (2016). Framework for big data usage in risk management process in banking institutions. Central European Conference on Information and Intelligent Systems, International Conference 2016 / Hunjak, Tihomir ; Kirinić, Valentina ; Konecki, Mario.