**贷后客户评分**

**版本：V1.1**

**目录**

[1 项目名称 3](#_Toc480993534)

[2 项目开发者 3](#_Toc480993535)

[3 项目开发技术及运行环境 3](#_Toc480993536)

[4 贷后评分目的 3](#_Toc480993539)

[5 贷后评分 方案功能实现 4](#_Toc480993540)

[5.1 数据获取 4](#_Toc480993541)

[5.1.1 获取合同粒度基础信息 4](#_Toc480993542)

[5.1.2 获取合同逾期情况 5](#_Toc480993543)

[5.1.3 总体数据集及输出 6](#_Toc480993544)

[5.2 定义贷后评分体系 7](#_Toc480993545)

[5.3 建立打分卡 8](#_Toc480993546)

[5.4 将数据远程传入业务方mysql数据库 10](#_Toc480993547)

# 项目名称

名称：贷后客户评分

# 项目开发者

王立立

# 项目开发技术及运行环境

开发技术：python，SQL

运行环境： linux ,python2.7 ,hive



# 贷后评分目的

贷后评分，旨在利用借款人贷后的特征，对贷后的表现进行排序，排序的结果以分值表现，其主要目的如下：

1、对现有的贷后状态进行描述，数值化的展现贷后表现；

2、利用贷后数据的统计含义，对未来的表现做出预估；

为此，我们的方案分为以下几步：

1、获取贷后相关数据特征及逾期情况；

2、定义贷后评分体系；

3、建立打分卡；

4、测试验证并将数据远程传入业务方mysql数据库。

# 贷后评分 方案功能实现

## 数据获取

### 获取合同粒度基础信息

以合同粒度抽取合同相关的基础信息；相关SQL见附件ml\_zhxd\_dh\_score\_basis.sql

抽取具体字段如下：

|  |  |
| --- | --- |
| 字段 | 字段含义 |
| apply\_id | 进件编号 |
| contract\_no | 合同号 |
| id\_number | 证件号码 |
| contract\_start\_date | 合同开始日期 |
| contract\_end\_date | 合同结束日期 |
| instalment | 当前所属期数 |
| contract\_amount | 合同金额 |
| issue\_amount | 放款金额 |
| cur\_loan\_balance\_all | 当前贷款余额(M7合同不为0) |
| loan\_term | 放款期数 |
| mn | 逾期期数 |
| early\_target | 提前结清 |
| is\_write\_off | 是否核销合同 |
| is\_first\_overdue | 首期逾期标识 |
| first\_term\_remain\_total | 首期逾期金额 |
| each\_term\_repayment\_start\_date | 每期应还日期 |
| each\_term\_business\_date | 每期实际还款日期 |
| each\_term\_overdue\_amt | 每期逾期金额 |

备注：

* 这样以合同为维度抽取合同的基本信息；主要对还款计划以合同为维度聚合，并保留每期 实还日期；应还日期；逾期金额；以”:”为分隔符保留。
* 上线时抽取时间应为数据库更新时间，即为当前时间的前一天：date\_sub(from\_unixtime(unix\_timestamp(),'yyyy-MM-dd'),1)。
* 产品范围：精英贷、新薪贷、新薪宜楼贷、精英贷（银行合作）、新薪贷（银行合作）。
* 提前结清：early\_target 逻辑，就是正常的提前结清，比如10期的合同，还到第5期就结清了; 取数逻辑：当还款日期为NULL；且应还所有费用之和=0 此合同此时表示提前结清。
* 计算当期期数：通过合同号与合同表关联，找到还款计划 最大的还款时间为最近一次还款时间，最大一次还款期数，为当前期数

### 获取合同逾期情况

根据以合同为维度的还款相关信息，根据每期还款计划，可得到实还日期及应还日期，这样可以得到每期还款计划的实际逾期，从而可以得到此合同每期还款计划的逾期情况M0-M7；具体实现见附件；overdue.py

|  |  |
| --- | --- |
| M0 | 逾期0天 |
| M1 | 逾期[1 -29] |

…. 顺序延后30天 为Mi,>M7的计为M7.

备注：

* 逾期计算：

days\_delta = （实还时间-应还时间）的天数

level = (days\_delta - 1) / 30 + 1

当level==1 and days\_delta<1时为 M0,其它情况都为level

* 当合同为核销 即有还款计划至今未还款，此时实还日期为NULL，此时我们将实还日期定 为程序所跑时间的前一天，因为当天数据跑出结果是前一天的还款计划情况。

比如合同10000501665 给出在4月25日跑出结果 我们得到部分还款计划及逾期情况如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 还款计划期数 | 逾期情况level | 应还日期 | 实还日期 | 当期未还款金额 |
| 1 | 1 | 2015-12-02 00:00:00 | 2015-12-07 00:00:00 | 1830.4 |
| 2 | 7 | 2016-01-02 00:00:00 | 2017-04-22 00:00:00 | 14504.76 |
| 3 | 7 | 2016-02-02 00:00:00 | 2017-04-22 00:00:00 | 13328.08 |
| 4 | 7 | 2016-03-02 00:00:00 | 2017-04-22 00:00:00 | 12249.82 |
| 5 | 7 | 2016-04-02 00:00:00 | 2017-04-22 00:00:00 | 11166.98 |
| 6 | 7 | 2016-05-02 00:00:00 | 2017-04-22 00:00:00 | 10155.64 |
| 7 | 7 | 2016-06-02 00:00:00 | 2017-04-22 00:00:00 | 9167.37 |
| 8 | 7 | 2016-07-02 00:00:00 | 2017-04-22 00:00:00 | 8249.1 |
| 9 | 7 | 2016-08-02 00:00:00 | 2017-04-22 00:00:00 | 7355.44 |
| 10 | 7 | 2016-09-02 00:00:00 | 2017-04-22 00:00:00 | 6509.45 |
| 11 | 7 | 2016-10-02 00:00:00 | 2017-04-22 00:00:00 | 5731.14 |
| 12 | 6 | 2016-11-02 00:00:00 | 2017-04-22 00:00:00 | 4979.77 |

由此份还款计划可知，从第二期开始之后都没有进行还款，初步判定为核销合同即M7。

### 总体数据集及输出

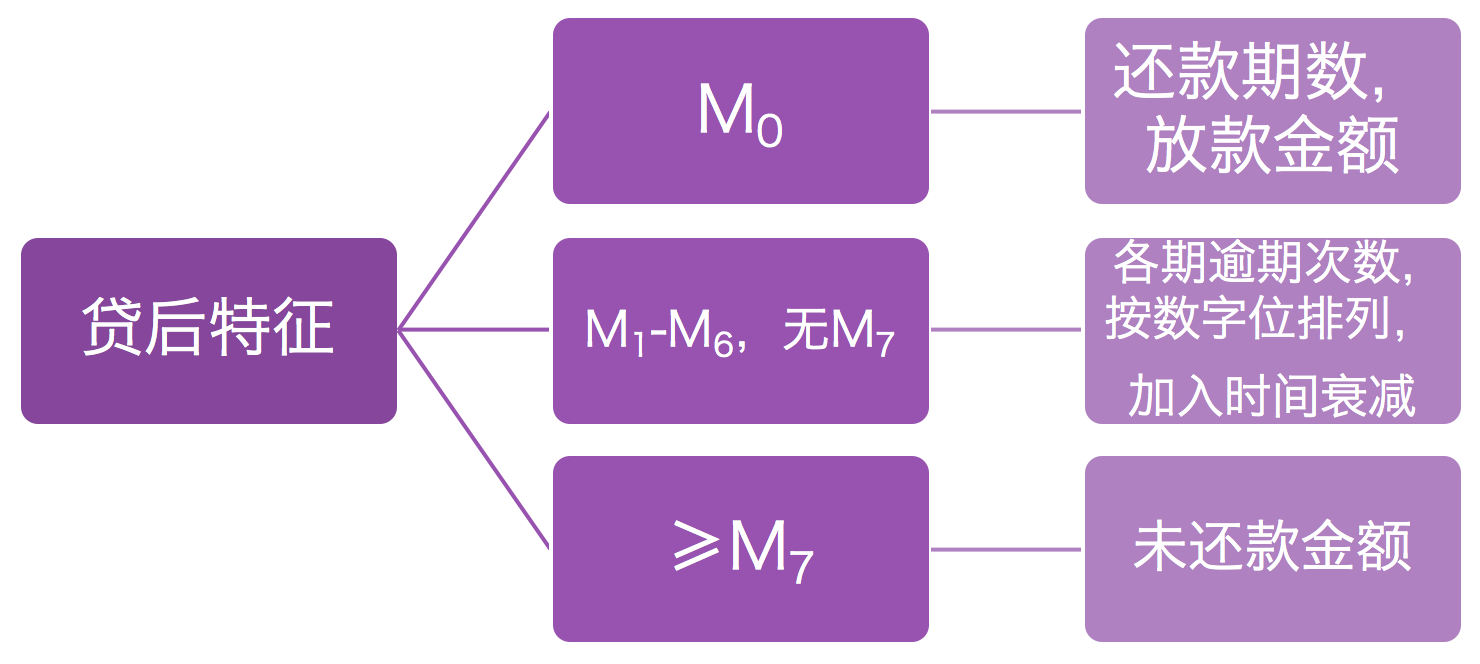
总体评分数据集：

获取至上个星期日之前的所有合同，并按照身份证聚合，用于生成总体评分。

输出值：以身份证为维度,得到评分值、排名百分数、档位、档位内排名百分数

## 定义贷后评分体系

首先，我们要对贷后数据建立评分体系（排序规则）。其过程如同人工构造一棵决策树模型：



根据提取的贷后特征及每期逾期情况，可以判断合同的逾期情况，即M0~M7

在级别内部根据贷后特征进行排序。具体排序情况如下

* M0 ：首先根据还款期数进行排序，在还款期数相同的情况下，对放款金额排序。
* M1-M6：对逾期次数进行排序，并且逾期次数加入衰减遗忘模型，按照数字位排序。
* >=M7：根据未还款金额排序。

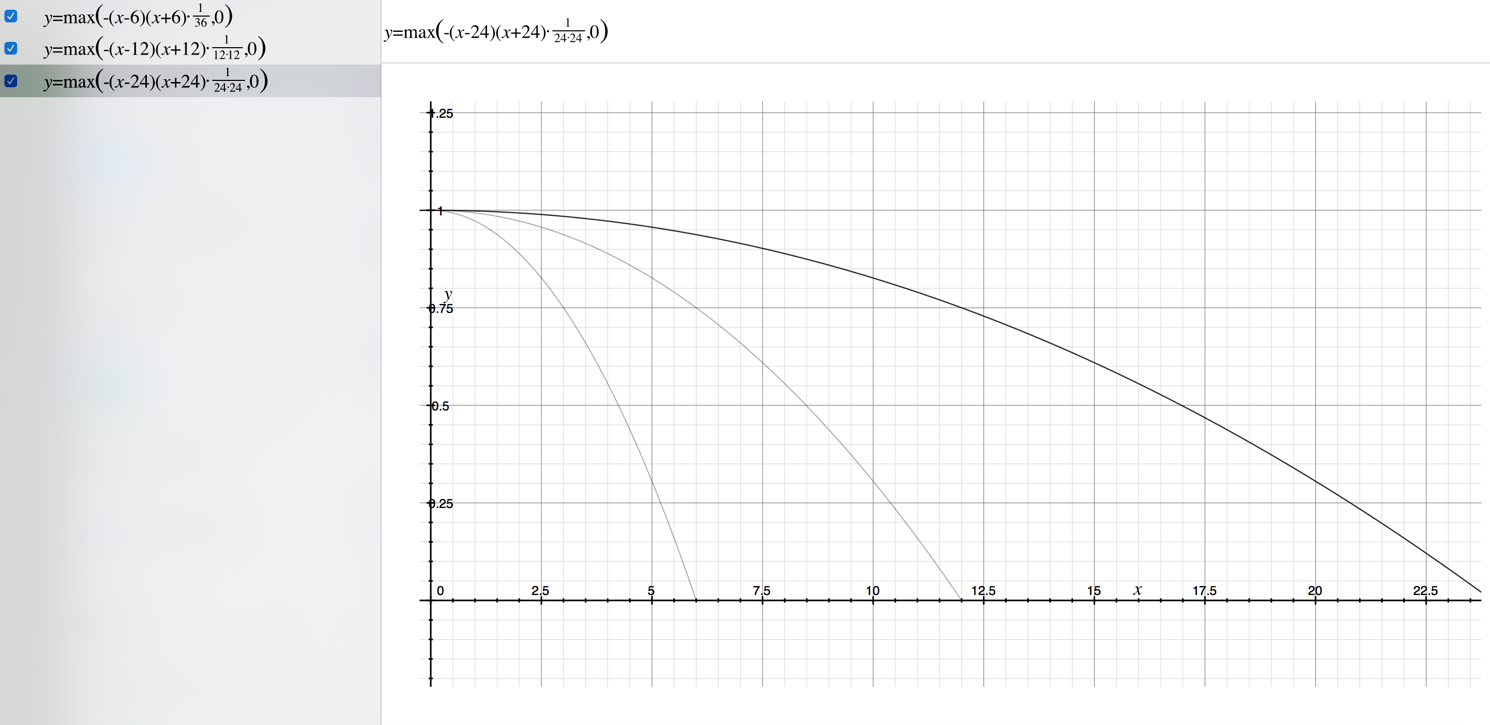
衰减遗忘模型：

由于客户表现有时好有时坏，要求评分可以动态变化。例如，某客户有过一次M3，但是后续很多期还款都小于M3，那么他的评分会逐渐升高。所有我们这面加入衰减遗忘模型

规则：客户每履行一期还款义务，则在此之前的还款期数的权重都衰减一次；如果客户没有发生新的还款，则不能执行衰减。因此衰减是相对每期还款行为的，不是相对自然时间流逝的。

   M0~M6都可以执行衰减，它们的初始权重系数都是1，衰减后<=1。

  衰减曲线设计：

 附：衰减曲线例子，假设当前期数是 t = 0，权重为1，之前的期数依次是 -1、-2、...、-N，其权重依次递减，当期数 t <= -N 时，都已经遗忘，权重都为0。图中假设 N ＝ 24（24期时间窗口），权重随期数增加呈二次衰减。

## 建立打分卡

**具体打分卡实现：**

* 以身份证为维度聚合合同
* 单笔合同处理

a:得到单笔合同的逾期Mi（逾期最大的逾期情况）

b: 非M7:M7\_cur\_loan\_balance\_all 当前贷款余额设置为0.0 M7:M7\_cur\_loan\_balance\_all为此笔合同的当前贷款余额

* 合同总金额，当前贷款余额，已还期数：为几个分合同的加和
* 对M1\_M6 进行衰减：

a:对几笔合同按照每期还款时间进行降续排序，将日期作为字典的key值， value对应其衰减输入期数，如果还款时间相同，则其应该有相同的衰减期数。

b:只有M1至M6直接才会进行衰减 并进行加和,得到降续排列衰减[M6,M5,M4，M3,M2,M1]

* 人为维度的合同逾期情况: 找到合同聚合之后的最大逾期情况， 如果最大逾期<0（提前还款）， 则最大逾期=0
* 排序：[max\_overdue\_term, M7\_cur\_loan\_balance\_all，M6,M5,M4,M3,M2,M1,-Return\_period(已还期数的负数，由于降续排序，已还期数越大应该分数越大)，-issue\_amount(合同金额，同理)] 进行降续排序。

具体实现见附件脚本 :dh\_score\_time.py

依照评分体系，对全体数据集,并把相应的序数和最大逾期期数放缩至0-1000分段，获得评分。

|  |  |
| --- | --- |
| **M\*** | **分值** |
| **M0** | **800-1000** |
| **M1** | **700-800** |
| **M2** | **600-700** |
| **M3** | **500-600** |
| **M4** | **400-500** |
| **M5** | **300-400** |
| **M6** | **200-300** |
| **≥M7** | **0-200** |

## 将数据远程传入业务方mysql数据库

由于对方数据库为mysql数据库，通过sqoop 命令可以远程将数据导入到业务方数据库。具体code见附件：sqoop\_shell.sh