



廣東財經大學  
GUANGDONG UNIVERSITY OF FINANCE & ECONOMICS

# 实验报告

课 程 名 称	云计算体系架构
实 验 项 目 名 称	预测德国信贷申请人的信用风险
班级与班级代码	16 计算机 2 班, 162511022
专 业	计算机科学与技术
任 课 教 师	谢嵘
学 号	16251102256
实 验 室 名 称	实验楼-801
姓 名	李胜欣
实 验 日 期	2019 年 05 月 29 日



# 预测德国信贷申请人的信用风险的分析

## 目录

预测德国信贷申请人的信用风险的分析 .....	1
第○部分：定量分析简介 .....	1
一、分析场景 .....	1
二、数据源 .....	1
三、贷款申请者信息 .....	1
四、要求 .....	2
五、开发环境 .....	2
第一部分：探索性数据分析，损失函数和初始模型拟合 .....	2
一、加载包 .....	3
二、加载数据 .....	3
三、可变分类 .....	5
四、检索响应变量 .....	6
五、二元变量 .....	7
六、非二进制分类变量 .....	8
八、损失函数 .....	11
(1) 无信息模型 (No-Information Model) .....	11
(2) 完美模型 (Perfect Model) .....	11
(3) 盲比例模型 (Blind Proportional Model) .....	12
九、探索初始模型空间 .....	12
第二部分：探索规范化模型 .....	14
第三部分：预测概率截止阈值优化 .....	16
结论：最终模型选择和指标 .....	19
第四部分：确定最相关的变量 .....	20
第五部分：最终结论 .....	22

# 第〇部分：定量分析简介

## 一、分析场景

一位贷款经理要求一个统计模型来帮助她的部门确定哪些贷款申请人是可信的，即最有可能偿还他们的贷款。在决定贷款申请之前，贷款经理会考虑申请人的人口统计和社会经济概况。经理的目标是最大限度地降低银行贷款组合的风险并实现利润最大化。经理分享有关模型预测的贷款类型的信息，如果借款人偿还贷款，银行将获得贷款价值的 35% 的利润。另一方面，如果借款人违约，银行的损失是 100%。对于被拒绝的申请人，银行不会赔钱，而且经理声称该模型不必考虑那些本应偿还贷款但被拒绝的申请人的机会成本。

在收到此请求后，我决定为经理开发一个模型，根据提供的数据最大化利润成本函数。在这种情况下，模型拟合任务的优先级将是预测，因为管理者没有特别请求可解释的模型，但是已经请求具有最佳利润特征的模型。

## 二、数据源

贷款经理让您可以访问 1000 个申请人的部门贷款数据样本，其中包括贷款结果。她声称数据集是由另一位分析师准备的，她的意见是代表银行的实际客户。

该期 末 作 业 的 数 据 源 由 谢 嵘 老 师 提 供 的 网 站 中 的 **German Credit data** - [german\\_credit.csv](#) 链接即可以获得，采用的是 CSV 格式。

## 三、贷款申请者信息特征

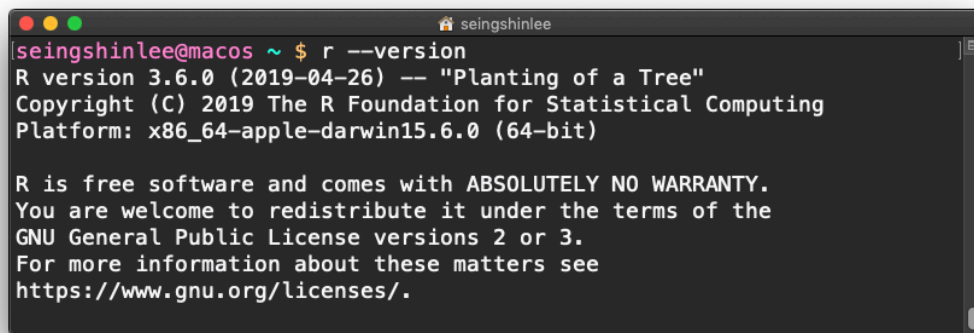
Feature	Description	Example
Creditability	label	0,1
Balance	Account Balance, Categorical	1
Duration	Duration of Credit in Months	18
History	History of Previous Credit, Categorical	4
Purpose	Purpose of Credit	0
Credit Amount	Numerical	1049.00
Savings	Value in Savings or Stocks Categorical	1
Employment	Length of current employment Categorical	2
instPercent	Installment Percent Categorical	1
sexMarried	Sex and Marital status categorical	1
guarantors	Guarantors categorical	0
Residence duration	Duration in Current address Categorical	1
assets	Most valuable available asset Categorical	2
age	Age (years) Numeric	20
concCredit	Concurrent Credits Categorical	1
apartment	Type of apartment Categorical	2
credits	No of Credits at this Bank, Numeric	1
occupation	categorical	1
dependents	No of dependents, Numeric	2
has Phone	Has phone categorical	1
foreign	Foreign worker	1

## 四、要求

在这个场景下，我们会构建一个由决策树组成的随机森林模型来预测是否守信用的标签/类别，基于以下特征：

- 1、标签 -> 守信用或者不守信用（1 或者 0）
- 2、特征 -> {存款余额，信用历史，贷款目的，信用额度，储蓄，已婚情况，分期存款百分比，年龄等等}

## 五、开发环境

A terminal window titled 'seingshinlee' on a macOS system. The prompt is 'seingshinlee@macos ~ \$'. The command 'r --version' has been executed, resulting in the following output: 'R version 3.6.0 (2019-04-26) -- "Planting of a Tree"', 'Copyright (C) 2019 The R Foundation for Statistical Computing', 'Platform: x86\_64-apple-darwin15.6.0 (64-bit)', 'R is free software and comes with ABSOLUTELY NO WARRANTY.', 'You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3.', 'For more information about these matters see https://www.gnu.org/licenses/'.

```
seingshinlee@macos ~ $ r --version
R version 3.6.0 (2019-04-26) -- "Planting of a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under the terms of the
GNU General Public License versions 2 or 3.
For more information about these matters see
https://www.gnu.org/licenses/.
```

## 第一部分：探索性数据分析，损失函数和初始模型拟合

首先，必须将数据加载到会话中并检查变量以确定其适当的类型。然后，必须检查每个预测变量相对于响应变量的方差，以确保不存在零方差预测变量。必须将利润/成本信息编程为可用于评估拟合模型的功能形式。最后，一组初始模型适合数据，以确定如何最好地进行。

## 一、加载包

```
1 # 在R语言中安装本题所需的包依赖
2 install.packages(data.table) # data.frame的扩展，快速聚合大数据
3 install.packages(gmodels) # 用于模型拟合的各种R编程工具
4 install.packages(DT) # 用于创建交互式的数据表
5 install.packages(gridExtra) # 提供许多用户级功能来处理“网格”图形，绘制表格
6 install.packages(pander) # 一个R的Pandoc Writer
7 install.packages(stringr) # 一个字符串处理工具集
8 install.packages(woe) # 计算证据和信息价值的权重
9 install.packages(MASS) # 支持Venables和Ripley的MASS的功能和数据集
10 install.packages(randomForest) # 布莱曼和卡特勒的随机森林分类和回归
11 install.packages(xgboost) # 是Gradient Boosting Machine的一个C++实现
12 install.packages(caret) # 一个用于解决分类和回归问题的数据训练综合工具包
13 install.packages(tidyverse) # tidyverse是专为数据科学设计的R包的集合
14 install.packages(RColorBrewer) # 绘图的柱形颜色的工具包
15
16 # 导入安装后所需的包依赖
17 library(data.table)
18 library(gmodels)
19 library(DT)
20 library(gridExtra)
21 library(pander)
22 library(stringr)
23 library(woe)
24 library(MASS)
25 library(randomForest)
26 library(xgboost)
27 library(caret)
28 library(tidyverse)
29 library(RColorBrewer)
30
31 # 导入自定义函数
32 if(!exists("createCVFolds", mode="function")) source("/Users/seingshinlee/云计算体系架构/createCVFolds.R")
33 if(!exists("get_best_cutoffs", mode="function")) source("/Users/seingshinlee/云计算体系架构/cutoff_check.R")
```

## 二、加载数据

```
1 # 使用data.table::fread将数据加载到tibble数据帧
2 credit = fread("german_credit.csv") %>%
3   tbl_df
4
5 # 将所有变量名称全部改为小写，删除无效字符，并用下划线
6 # 替换空格，将所有变量名称转换为snake_case
7 names(credit) = names(credit) %>%
8   tolower %>%
9   str_replace_all("[ /]", "_") %>%
10  str_replace_all("[(&)]", "")
11
12 # 显示预览
13 credit %>% datatable(style="bootstrap")
```

许多变量名称无效，例如“信用期（月）”。空格和无效括号字符可能会在编程使用这些预测变量时出现问题。

```

1 # 使用data.table::fread将数据加载到tibble数据帧
2 credit = fread("german_credit.csv") %>%
3   tbl_df
4
5 # 将所有变量名称全部改为小写，删除无效字符，并用下划线
6 # 替换空格，将所有变量名称转换为snake_case
7 names(credit) = names(credit) %>%
8   tolower %>%
9   str_replace_all("[ /]", "_") %>%
10  str_replace_all("[(&)]", "")
11
12 # 显示预览
13 credit %>% datatable(style="bootstrap")
14
15 # 结果: german_credit.csv转换成ASCII表格
16 +-----+-----+-----+-----+-----+-----+-----+-----+
17 | | Account Balance | Duration of Credit (month) | Payment Status of Previous Credit | Purpose |
18 | Credit Amount | Value Savings/Stocks | Length of current employment | Instalment per cent |
19 | Sex & Marital Status | Guarantors | Duration in Current address | Most valuable available asset |
20 | Age (years) | Concurrent Credits | Type of apartment | No of Credits at this Bank | Occupation |
21 | No of dependents | Telephone | Foreign Worker |
22 +-----+-----+-----+-----+-----+-----+-----+-----+
23
24 .....
25
26 +-----+-----+-----+-----+-----+-----+-----+-----+
27 | 0 | 12680 | 4 | 5 | 21 | 4 | 0 |
28 | 30 | 1 | 2 | 1 | 3 | 1 | 4 |
29 | 0 | 6468 | 2 | 5 | 12 | 2 | 3 |
30 | 52 | 1 | 2 | 1 | 2 | 1 | 4 |
31 | 0 | 6350 | 1 | 5 | 30 | 2 | 2 |
32 | 31 | 1 | 1 | 1 | 2 | 1 | 3 |

```

注解：上图结果，因为屏幕宽度有限，不能一行显示全部，表格部分被折叠了。

### 三、可变分类

探索数据的第一步是确定数据中存在哪种类型的变量。变量是分类还是定量的？如果他们分类的，他们是二元的还是他们有多个级别？

```
1 # 强制所有变量到因子并计算存在的唯一级别数。
2 credit_types = credit %>%
3   mutate_all(factor) %>%
4   map(levels) %>%
5   map(length) %>%
6   tbl_df %>%
7   gather(variable, n_unique) %>%
8   arrange(n_unique) %>%
9   mutate(binary = n_unique==2, categorical = n_unique <=10, continuous = n_unique > 10)
10
11 # 预览数据
12 datatable(credit_types, style="bootstrap")
13
14 # credit_types的ASCII的输出结果
15 +-----+-----+-----+-----+-----+
16 | variable | n_unique | binary | categorical | continuous |
17 +-----+-----+-----+-----+-----+
18 | 8 | account_balance | 4 | False | True | False |
19 +-----+-----+-----+-----+-----+
20 | 20 | age_years | 53 | False | False | True |
21 +-----+-----+-----+-----+-----+
22 | 6 | concurrent_credit | 3 | False | False | False |
23 +-----+-----+-----+-----+-----+
24 | 21 | credit_amount | 923 | False | False | True |
25 +-----+-----+-----+-----+-----+
26 | 1 | creditability | 2 | True | True | False |
27 +-----+-----+-----+-----+-----+
28 | 11 | duration_in_current_address | 4 | False | True | False |
29 +-----+-----+-----+-----+-----+
30 | 19 | duration_of_credit_month | 33 | False | False | True |
31 +-----+-----+-----+-----+-----+
32 | 4 | foregin_worker | 2 | True | True | False |
33 +-----+-----+-----+-----+-----+
34 | 5 | guarantors | 3 | False | True | False |
35 +-----+-----+-----+-----+-----+
36 | 9 | instalment_per_cent | 4 | False | True | False |
37 +-----+-----+-----+-----+-----+
38
39 # 中间省略若干行结果
40 .....
41
42 +-----+-----+-----+-----+-----+
43 | 3 | telephone | 2 | True | True | False |
44 +-----+-----+-----+-----+-----+
45 | 7 | type_of_apartment | 2 | True | True | False |
46 +-----+-----+-----+-----+-----+
47 | 16 | value_savings_stocks | 5 | False | True | False |
48 +-----+-----+-----+-----+-----+
```

在这种情况下，具有十个或更少唯一级别的变量被视为分类变量。对于每个数据集，这种假设都不适用，因为某些分类因素将具有超过十个级别。但是，此截止值适用于此特定数据集，因为最大分类变量具有 10 个级别。

存在四个二元分类变量，包括响应变量 creditability。

存在十四个非二元分类变量。

存在三个定量变量。

现在，可以探索每种变量类型。



## 四、检索响应变量

首先，响应变量被单独考虑，因为必须确定响应变量的分布。

```
1 # 结果: ASCII表格表示
2 # 元内容
3 +-----+
4 |                                     N |
5 +-----+
6 | N / Table Total |
7 +-----+
8
9 # 表中的总观察值: 1000
10 +-----+
11 | 0 | 1 |
12 +-----+
13 | 300 | 700 |
14 +-----+
15 | 0.300 | 0.700 |
16 +-----+
```

creditability 有两个级别，在正类中有 700 个观测值，在负类中有 300 个观测值。如果数据代表银行的贷款申请人，大约 70% 的申请人可以偿还贷款，30% 的人不能支付。类别的不平衡用于建模，因为有更多的信息可用于对正类进行分类，这可能允许某些模型在预测正类时产生良好的准确性，但在预测负类时具有较差的准确性，同时具有良好的总体准确性。

当前类型的数据将类视为 1 或 0。让我们重构此变量，使其更易于解释。

```
1 # 分配正类1, 标签“好”和负类0, 标签“差”
2 credit$creditability = ifelse(credit$creditability == 1, "Good", "Poor")
3 credit$creditability = factor(credit$creditability, levels=c("Good", "Poor"))
4 # 检索响应变量的分布
5 CrossTable(credit$creditability)
6
7 # 结果: ASCII表格表示
8 # 元内容
9 +-----+
10 |                                     N |
11 +-----+
12 | N / Table Total |
13 +-----+
14
15 # 表中的总观察值: 1000
16 +-----+
17 | Good | Poor |
18 +-----+
19 | 700 | 300 |
20 +-----+
21 | 0.700 | 0.300 |
22 +-----+
```

## 五、二元变量

现在，分析剩余的三个二进制变量。

```
1 # 确定二进制，分类变量的名称
2 binary_names = credit_types %>%
3   filter(binary) %>%
4   .$variable
5
6 # 考虑这些变量
7 credit = credit %>% mutate_at(binary_names, factor)
```

在分析的这个阶段确定的最重要的信息是任何预测变量是否具有零方差。当预测器的所有值与响应相同时，它的方差为零。例如，如果预测变量的所有观察结果仅具有“良好”可信度。在这种情况下，预测器的值不区分响应中的两个类。许多模型将无法适应零方差预测因子。此外，如果预测器的任何单个类别在响应方面为零，则模型拟合将失败，因为许多模型将分类变量分离为单独的预测变量。如果任何预测变量类别在响应方面的实例为零，则模型将无法拟合。

有关响应的零实例的任何预测变量类别？

```
1 # 有关响应的零实例的任何预测变量类别
2 length(checkConditionalX(credit[,binary_names] %>% select(-creditability), credit$creditability)) > 0
3
4 # 结果:
5 [1] FALSE
```

```
1 # 确定是否有任何预测变量零方差
2 nzv = nearZeroVar(credit[,binary_names], names = TRUE, saveMetrics = TRUE) %>% select(-percentUnique)
3 indexes = rownames(nzv)
4 pandoc.table(data.frame(indexes = indexes, nzv) %>% tbl_df %>% arrange(desc(freqRatio)))
5
6 # 结果: ASCII表格表示
7 +-----+-----+-----+-----+
8 | indexes | freqRatio | zeroVar | nzv |
9 +-----+-----+-----+-----+
10 | foreign_worker | 26.03 | False | True |
11 +-----+-----+-----+-----+
12 | no_of_dependents | 5.452 | False | False |
13 +-----+-----+-----+-----+
14 | creditability | 2.333 | False | False |
15 +-----+-----+-----+-----+
16 | telephone | 1.475 | False | False |
17 +-----+-----+-----+-----+
```

这些二元预测变量中没有一个是具有零方差，但是 `foreign_worker` 具有接近零的方差，其中一个类别比另一个类别更普遍 26 倍 `creditability`。此预测变量将保留在数据中，但如果模型构建需要正则化，则这种接近于零的方差可能使其成为分析后期删除的可能候选者。

## 六、非二进制分类变量

接下来，检查非二进制分类变量。

```
1 # 标识非二进制，分类变量的名称
2 nb_names = credit_types %>% filter(!binary,categorical) %>% .$variable
3 # 考虑这些变量
4 credit = credit %>% mutate_at(nb_names, factor)
```

还检查这些预测变量的零方差。有关响应的零实例的任何预测变量类别？

```
1 # 有关响应的零实例的任何预测变量类别
2 length(checkConditionalX(credit[,nb_names], credit$creditability)) > 0
3 # 结果:
4 [1] FALSE
```

```
1 # 确定是否有任何预测变量零方差
2 nzv = nearZeroVar(credit[,nb_names], names = TRUE, saveMetrics = TRUE) %>% select(-percentUnique)
3 indexes = rownames(nzv)
4 pandoc.table(data.frame(indexes = indexes, nzv) %>% tbl_df %>% arrange(desc(freqRatio)))
5
6 # 结果: ASCII表格表示
7 +-----+-----+-----+-----+
8 |               indexes |   freqRatio | zeroVar |   nzv |
9 +-----+-----+-----+-----+
10 | guarantors            |    17.44    |   False |   True |
11 +-----+-----+-----+-----+
12 | concurrent_credits    |     5.856   |   False |   False |
13 +-----+-----+-----+-----+
14 | type_of_apartment     |     3.989   |   False |   False |
15 +-----+-----+-----+-----+
16 | value_savings_stocks  |     3.295   |   False |   False |
17 +-----+-----+-----+-----+
18 | occupation            |     3.15    |   False |   False |
19 +-----+-----+-----+-----+
20 | instalment_per_cent   |     2.061   |   False |   False |
21 +-----+-----+-----+-----+
22 | no_of_credits_at_this_bank |    1.901   |   False |   False |
23 +-----+-----+-----+-----+
24 | payment_status_of_previous_credit |    1.809   |   False |   False |
25 +-----+-----+-----+-----+
26 | sex_marital_status    |     1.768   |   False |   False |
27 +-----+-----+-----+-----+
28 | account_balance       |     1.438   |   False |   False |
29 +-----+-----+-----+-----+
30 | duration_in_current_address |    1.341   |   False |   False |
31 +-----+-----+-----+-----+
32 | length_of_current_employment |    1.34    |   False |   False |
33 +-----+-----+-----+-----+
34 | purpose               |     1.197   |   False |   False |
35 +-----+-----+-----+-----+
36 | most_valuable_available_asset |    1.177   |   False |   False |
37 +-----+-----+-----+-----+
```

虽然 guarantors 具有高频率比, 但这些预测器中没有一个被认为具有零方差或接近零方差。

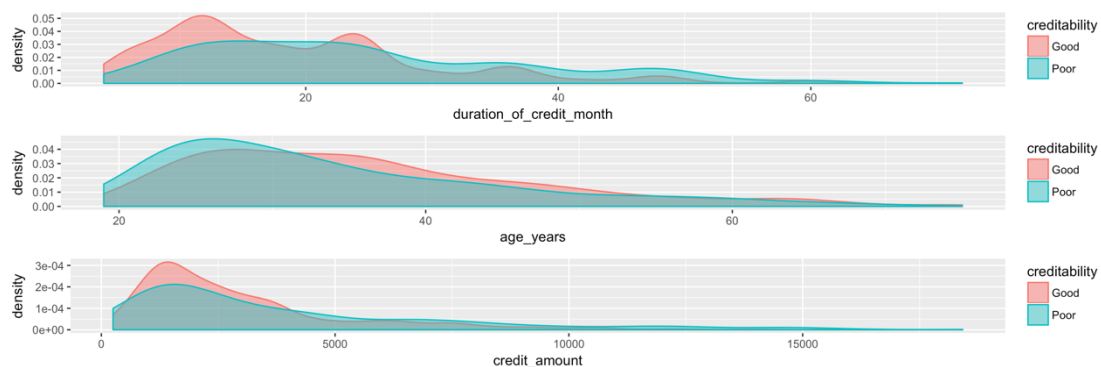
## 七、连续变量

最后，总结了连续变量。

```
1 # 标识非二进制，分类变量的名称
2 quant_names = credit_types %>% filter(continuous) %>% .$variable
3
4 # 辅助函数总结每个预测变量
5 quant_summary = function(data, vector_name) {
6   data %>% summarize_at(vector_name, funs(MIN=min, Q1 = quantile(., 0.25), MEAN = mean, MEDIAN = median,
7     Q3 = quantile(., 0.75), MAX = max, IQR = IQR, STDEV = sd)) %>%
8     mutate(SKEW = ifelse(MEAN > MEDIAN, "RIGHT", "LEFT"))
9 }
10
11 # 输出表
12 data.frame(Predictor = quant_names,
13   bind_rows(
14     quant_summary(credit, quant_names[1]),
15     quant_summary(credit, quant_names[2]),
16     quant_summary(credit, quant_names[3])) %>%
17   pandoc.table(split.tables=Inf))
18
19 # 结果: ASCII表格表示
20 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
21 | Predictor | MIN | Q1 | MEAN | MEDIAN | Q3 | MAX | IQR | STDEV | SKEW |
22 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
23 | duration_of_credit_month | 4 | 12 | 20.9 | 18 | 24 | 72 | 12 | 12.06 | RIGHT |
24 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
25 | age_years | 19 | 27 | 35.54 | 33 | 42 | 75 | 15 | 11.35 | RIGHT |
26 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
27 | credit_amount | 250 | 1366 | 3271 | 2320 | 3972 | 18424 | 2607 | 2823 | RIGHT |
28 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
29
```

所有三个变量都显示出明显的正偏度。预测因子的密度图更清楚地证明了这一点。

```
1 # 三个变量的预测因子的密度图
2 left = ggplot(credit, aes(duration_of_credit_month, fill=creditability, color=creditability)) +
3   geom_density(alpha=0.5)
4 middle = ggplot(credit, aes(age_years, fill=creditability, color = creditability)) +
5   geom_density(alpha=0.5)
6 center = ggplot(credit, aes(credit_amount, fill=creditability, color = creditability)) +
7   geom_density(alpha=0.5)
8 grid.arrange(left,middle,center, ncol=1)
```



## 八、损失函数

在对变量进行基本探索性分析之后，第一部分模型可以适合数据。但是，为了评估模型，必须实现利润 - 成本信息的功能化。

```
1 # Profit35根据预测的类值和实际的类值计算平均利润
2 Profit35 = function(actual, pred, positive=NULL) {
3   # 生成混淆矩阵
4   Confusion_DF <- MLmetrics::ConfusionDF(pred, actual)
5   if (is.null(positive) == TRUE)
6     positive <- as.character(Confusion_DF[1, 1])
7   # 确定True Positive率
8   TP <- as.integer(subset(Confusion_DF, y_true == positive &
9     y_pred == positive)["Freq"])
10  # 确定False Positive率
11  FP <- as.integer(sum(subset(Confusion_DF, y_true != positive &
12    y_pred == positive)["Freq"]))
13  # 计算平均利润
14  val_35 = (TP / sum(Confusion_DF$Freq))*0.35 + (FP / sum(Confusion_DF$Freq))*-1
15  return(val_35)
16 }
17 # p35是插入符号包的包装器，允许在模型评估结果中将Profit35值包含在特性和准确性中
18 p35 = function(data, lev = NULL, model = NULL) {
19   p35_val <- Profit35(data$obs, data$pred, lev[1])
20   spec = MLmetrics::Specificity(data$obs, data$pred)
21   acc = MLmetrics::Accuracy(data$pred, data$obs)
22   return(c(Accuracy = acc, Specificity= spec, P35 = p35_val))
23 }
```

可以使用来自无信息模型的假设预测来测试该成本函数，该无信息模型始终预测最常见的类“好”和完美模型，该模型预测所有类与数据中观察到的完全相同。

### (1) 无信息模型 (No-Information Model)

```
1 # 无信息模型
2 p35(data.frame(obs = credit$creditability, pred = factor(c(rep("Good", 1000)), levels =
3   levels(credit$creditability))))
4 # 结果:
5 Accuracy    Specificity    P35
6 0.700      0.000      -0.055
```

### (2) 完美模型 (Perfect Model)

```
1 # 完美模型
2 p35(data.frame(obs = credit$creditability, pred = credit$creditability))
3 # 结果:
4 Accuracy    Specificity    P35
5 1.000      1.000      0.245
```

无信息模型的准确度为 70%，特异性为 0%，P35 值为-0.055。因此，无论可信度如何，

总是发放贷款的虚拟模型（将所有申请人标记为“良好”）预计将有 0.055 单位损失。如果平均贷款金额为 10,000 美元，则使用此模型的总损失为 550,000 美元，每个申请人损失为 550 美元。请注意，特异性包括在内，因为特异性（1 - 假阳性率，或 TN / 总和（预测阴性））将与 P35 密切相关。存在这种关系是因为成本函数对误报有很大的惩罚，当模型预测申请人是“好”时，他们应该是“坏”。这个例子也表明了模型的准确度如何在 70% 时保持不错，但预期的单位利润是负的，亏损。

完美模型的准确度为 100%，特异度为 100%，P35 为 0.245。因此，如果存在一个完美的模型，它将获得 0.245 单位的利润。如果平均贷款金额为 10,000 美元，则使用此模型的总利润为 2,450,000 美元，每个申请人的利润为 2450 美元。在这种情况下，银行只能获利，因此，所有贷款的利润为 35%。重要的是，任何模型都不应超过这个假设值，任何接近它的模型都应该仔细评估，因为没有模型是完美的。

在没有其他信息的情况下，在此分析中用于比较的最相关的零模型是一个模型，表示随机分配“良好”或“差”可信度，假设贷款申请人在 70-30 比率下是可信的。因此，这种策略总是预测 700 名“好”和 300 名“差”申请人。但是，预测会随机分配给申请人。这种策略的性能会有所不同，具体取决于哪些申请人被随机分配到哪个类别。实际上，这种策略的可能性之一是所有预测都是 100% 准确的。然而，这种情况发生的可能性很小。考虑到这种随机变化，为了捕获模型的性能，使用了模型的平均性能。

### （3）盲比例模型（Blind Proportional Model）

```
1 # 盲比例模型
2 blind = credit %>%
3   select(creditability) %>%
4   mutate(row_num = 1:1000) %>%
5   arrange(creditability) %>%
6   mutate(predicted = c(rep("Good", 490), rep("Poor", 210), rep("Good", 210), rep("Poor", 90))) %>%
7   arrange(row_num)
8
9 p35(data.frame(obs = blind$creditability, pred = blind$predicted))
10
11 # 结果:
12 Accuracy    Specificity    P35
13 0.5800      0.3000      -0.0385
```

盲比例模型的准确度为 58%，特异性为 30%，P35 值为 -0.039。因此，平均而言，这种策略比无信息模型提高了大约 0.016，但仍然代表了银行的平均损失。

## 九、探索初始模型空间

由于预测质量是此问题的优先级，因此会立即评估一系列不同的模型类型，以查看是否有任何特定模型产生更好的预测。包含的模型是逻辑回归，线性判别分析，随机森林和 xgboost 梯度提升树。

随机森林和 xgboost 都有可以调整的超参数。以下代码块包括设置为其最佳值的超参数。

将训练模型以使用该 p35 功能最大化每单位利润。为了更准确地估算每个模型的每单位利润，使用 10x10 K 折叠交叉验证（CV）过程。由于正在比较多个模型，因此每个模型将在同一组折叠上进行训练。

```

1 # 探索初始模型空间
2 set.seed(123)
3 repeatedResamples = function(y, k = 10, reps=10) {
4   suppressWarnings(
5     for (idx in 1:reps) {
6       # 创建自定义索引: myFolds
7       myFolds <- createCVFolds(y, k = k)
8
9       # 创建可重用的trainControl对象: myControl
10      myControl <- trainControl(summaryFunction = p35,
11                               classProbs = TRUE, # 重要!
12                               verboseIter = FALSE,
13                               savePredictions = TRUE,
14                               index = myFolds
15                               )
16
17      log_fit = train(creditability ~ .,
18                    method = 'glm',
19                    family = 'binomial',
20                    data = credit,
21                    trControl = myControl,
22                    metric = "P35")
23
24      lda_fit = train(creditability ~ .,
25                    method = 'lda',
26                    show = FALSE,
27                    data = credit,
28                    trControl = myControl,
29                    metric = "P35")
30
31      tuneGridRF = data.frame(
32        mtry=11
33      )
34
35      rf_fit = train(creditability ~ .,
36                   method = 'rf',
37                   data = credit,
38                   trControl=myControl,
39                   tuneGrid=tuneGridRF,
40                   metric = "P35")
41
42      tuneGridXGB <- expand.grid(
43        nrounds=300,
44        max_depth = 2,
45        eta = 0.07,
46        gamma = 0.1,
47        colsample_bytree = 1,
48        subsample = 1,
49        min_child_weight = 2)
50
51      xgb_fit = train(creditability ~ .,
52                    method = 'xgbTree',
53                    data = credit,
54                    trControl= myControl,
55                    tuneGrid = tuneGridXGB,
56                    metric = "P35")
57
58      # 创建model_list
59      model_list <- list(log = log_fit, lda = lda_fit, rf = rf_fit, xgb = xgb_fit)
60
61      if (idx == 1) {
62        # 将model_list传递给resamples(): resamples
63        resamples <- resamples(model_list)
64      }
65      else {
66        current_resample = resamples(model_list)
67        resamples$values = bind_rows(resamples$values, current_resample$values)
68      }
69    }
70  }
71 }
72 return(resamples)
73 }
74 }
75 report_dat = repeatedResamples(credit$creditability)
76 summary(report_dat)

```



```
1 # 结果:
2 Call:
3 summary.resamples(object = fit_results)
4
5 Models: log_reg, rlda
6 Number of resamples: 100
7
8 Accuracy
9      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA(s)
10 log_reg 0.68  0.7300  0.76 0.7558  0.78 0.84  0
11 rlda    0.58  0.6475  0.68 0.6779  0.71 0.79  0
12
13 P35
14      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA(s)
15 log_reg 0.0035  0.04 0.05675 0.055075 0.071375 0.1435  0
16 rlda    0.0315  0.07 0.08750 0.087935 0.104500 0.1510  0
17
18 Specificity
19      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA(s)
20 log_reg 0.3000000 0.4000000 0.4666667 0.4643333 0.5 0.7666667  0
21 rlda    0.5333333 0.7333333 0.7666667 0.7726667 0.8 0.9666667  0
```

总的来说，这四种模型比盲目模型的性能有所提高。每个模型在 CV 折叠上的性能指标的中位数用于度量估计，因为中位数不依赖于性能指标分布的尾部行为，因此对  $k$ -期间创建的异常数据点具有高度抵抗力。折叠过程。

这一轮的最佳模型是 `xgboost` 模型。

将正则化引入模型拟合程序可能是适当的。正则化引入了对模型复杂性的惩罚，以减少过度拟合。过度拟合会导致样本外预测的模型方差增加，从而降低模型质量和准确性。由于许多变量相对于其他变量具有较低的平均增益，因此模型可能会通过尝试从与响应几乎没有关系的变量绘制关系来在一定程度上过度拟合数据。像 `xgboost` 和随机森林这样的模型对这个问题更加健壮。但是，逻辑回归和线性判别分析对这类问题并不稳健，可能会从正规化中得到改善。

## 第二部分：探索规范化模型

对于第二部分，目标是在不涉及树的模型中使用正则化。方法“`regLogistic`”将用于正则化逻辑回归，“`rlda`”用于正则化线性判别分析。

```

1 set.seed(123)
2 repeatedResamples = function(y, data, k = 10, reps=10) {
3   suppressWarnings(
4     for (idx in 1:reps) {
5       # 创建自定义的indices: myFolds
6       myFolds <- createCVFolds(y, k = k)
7
8       # 创建可重用的trainControl object: myControl
9       myControl <- trainControl(summaryFunction = p35,
10                                classProbs = TRUE, # 重要!
11                                verboseIter = FALSE,
12                                savePredictions = TRUE,
13                                index = myFolds
14                                )
15
16       round2_tuneGridRegLog = expand.grid(
17         cost = 1,
18         loss = "L1",
19         epsilon = 0.01
20       )
21
22       log_reg_fit = train(creditability ~ .,
23                          method = 'regLogistic',
24                          data = data,
25                          trControl=myControl,
26                          tuneGrid = round2_tuneGridRegLog,
27                          metric = "P35")
28
29       round2_tuneGridrlda = data.frame(estimator = "Moore-Penrose Pseudo-Inverse")
30
31       rlda_fit = train(creditability ~ .,
32                       method = 'rlda',
33                       data = data,
34                       trControl=myControl,
35                       tuneGrid = round2_tuneGridrlda,
36                       metric = "P35")
37
38       # 创建model_list
39       model_list <- list(log_reg = log_reg_fit, rlda = rlda_fit)
40
41       if (idx == 1) {
42         # 将model_list传递给resamples(): resamples
43         resamples <- resamples(model_list)
44       }
45       else {
46         current_resample = resamples(model_list)
47         resamples$values = bind_rows(resamples$values, current_resample$values)
48       }
49     }
50   )
51   return(resamples)
52 }
53
54 }
55
56 fit_results = repeatedResamples(credit$creditability, credit)
57 summary(fit_results)

```

```

1 # 结果:
2 Call:
3 summary.resamples(object = fit_results)
4
5 Models: log_reg, rlda
6 Number of resamples: 100
7
8 Accuracy
9      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA(s)
10 log_reg 0.68  0.7300  0.76 0.7558   0.78 0.84    0
11 rlda    0.58  0.6475  0.68 0.6779   0.71 0.79    0
12
13 P35
14      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA(s)
15 log_reg 0.0035   0.04 0.05675 0.055075 0.071375 0.1435    0
16 rlda    0.0315   0.07 0.08750 0.087935 0.104500 0.1510    0
17
18 Specificity
19      Min. 1st Qu. Median   Mean 3rd Qu. Max. NA(s)
20 log_reg 0.3000000 0.4000000 0.4666667 0.4643333   0.5 0.7666667   0
21 rlda    0.5333333 0.7333333 0.7666667 0.7726667   0.8 0.9666667   0

```

正则化逻辑回归的表现类似于逻辑回归，正则化逻辑回归的单位利润中位数为 0.057，逻辑回归为 0.055。将继续使用原始逻辑回归。

另一方面，正则化线性判别分析得出单位利润为 0.088，而线性判别分析的单位利润为 0.054。“rlda”模型的显着表现与其 0.77 的高特异性相关，导致较低的误报率，这意味着较少的信用不良申请人被归类为具有良好信用的模型。

### 第三部分：预测概率截止阈值优化

迄今为止的分析假设之一是用于确定预测是属于正类还是负类的截止阈值是 0.5。该值意味着如果模型给出大于 0.5 的预测概率，则预测将是“好”，反之亦然。该阈值改变了模型性能的灵敏度和特异性。由于利润 - 成本函数对模型的特异性敏感，这与其假阳性率成反比，下一步将是修改最有希望的模型的预测截止概率以支持更高的特异性。实际上，如果预测概率截止值增加到 0.5 以上，则该模型将不太可能预测“良好”可信度，

此步骤的功能可在 cutoff\_check.R 文件中找到，并涉及 10x10 K-fold 交叉验证，类似于此分析中先前使用的验证程序。该过程将截止值评估为 0.05 (0.50, 0.55, 0.6 等) 差异。

```

1 # 使用10x10的K折叠交叉验证确定最佳截止值
2 best_cutoff = get_best_cutoffs(formula = creditability ~ ., data = credit, method = "glm",
3 family="binomial", verbose = FALSE)
3 best_cutoff %>% pandoc.table()

```

```

1 # 使用10x10的K折叠交叉验证确定最佳截止值
2 best_cutoff = get_best_cutoffs(formula = creditability ~ ., data = credit, method = "glm",
  family="binomial", verbose = FALSE)
3 best_cutoff %>% pandoc.table()
4
5 # 结果: ASCII表格表示
6 +-----+-----+-----+
7 | method | best_cutoff | best_metric_median |
8 +-----+-----+-----+
9 | glm | 0.7 | 0.08825 |
10 +-----+-----+-----+

```

```

1 # 保存最终结果以与其他模型进行比较
2 final_results = train_cv(formula = creditability ~ ., data = credit, , method = "glm", family="binomial",
  verbose=FALSE, cutoff = 0.7)
3 log_perf = data.frame(value = final_results$P35, group="Logistic")

```

从逻辑回归模型开始，最佳截止值为 0.7，这将模型的单位利润估计值从 0.057 增加到 0.088。

```

1 # 使用10x10的K折叠交叉验证确定最佳截止值
2 round1_tuneGridRF = data.frame(mtry=11)
3 best_cutoff = get_best_cutoffs(formula = creditability ~ ., data = credit, method = "rf", tuneGrid =
  round1_tuneGridRF, verbose=FALSE)
4 best_cutoff %>% pandoc.table()
5
6 # 结果: ASCII表格表示
7 +-----+-----+-----+
8 | method | best_cutoff | best_metric_median |
9 +-----+-----+-----+
10 | rf | 0.7 | 0.0865 |
11 +-----+-----+-----+

```

```

1 # 保存最终结果以与其他模型进行比较
2 final_results = train_cv(formula = creditability ~ ., data = credit, , method = "rf", tuneGrid =
  round1_tuneGridRF, verbose=FALSE, cutoff = 0.7)
3 rf_perf = data.frame(value = final_results$P35, group="Random Forest")

```

随机森林模型的单位利润估计值从 0.034 提高到 0.087，截止值为 0.7。

```

1 # 使用10x10的K折叠交叉验证确定最佳截止值
2 round1_tuneGridXGB <- expand.grid(
3     nrounds=300,
4     max_depth = 2,
5     eta = 0.07,
6     gamma = 0.1,
7     colsample_bytree = 1,
8     subsample = 1,
9     min_child_weight = 2)
10
11 best_cutoff = get_best_cutoffs(formula = creditability ~ ., data = credit, method = "xgbTree", tuneGrid =
round1_tuneGridXGB, verbose=FALSE)
12 best_cutoff %>% pandoc.table()
13
14 # 结果: ASCII表格表示
15 +-----+-----+-----+
16 | method | best_cutoff | best_metric_median |
17 +-----+-----+-----+
18 | xgbTree | 0.7 | 0.092 |
19 +-----+-----+-----+

```

```

1 # 保存最终结果以与其他模型进行比较
2 final_results = train_cv(formula = creditability ~ ., data = credit, method = "xgbTree",
tuneGrid=round1_tuneGridXGB, verbose=FALSE, cutoff = 0.7)
3 xgb_perf = data.frame(value = final_results$P35, group="Xgboost")

```

```

1 # 使用10x10的K折叠交叉验证确定最佳截止值
2 round2_tuneGridrlda = data.frame(estimator = "Moore-Penrose Pseudo-Inverse")
3 best_cutoff = get_best_cutoffs(formula = creditability ~ ., data = credit, method = "rlda",
tuneGrid=round2_tuneGridrlda, verbose=FALSE)
4 best_cutoff %>% pandoc.table()
5
6 # 结果: ASCII表格表示
7 +-----+-----+-----+
8 | method | best_cutoff | best_metric_median |
9 +-----+-----+-----+
10 | rlda | 0.5 | 0.092 |
11 +-----+-----+-----+

```

```

1 # 保存最终结果以与其他模型进行比较
2 final_results = train_cv(formula = creditability ~ ., data = credit, method = "rlda",
tuneGrid=round2_tuneGridrlda, verbose=FALSE, cutoff = 0.7)
3 rlda_perf = data.frame(value = final_results$P35, group="Regularized LDA")

```

使用 0.7 的截止值，xgboost 模型的单位利润估计值从 0.054 提高到 0.092。这个结果是迄今为止看到的最佳中位数 CV 表现。

使用 0.5 的截止值，正则化 LDA 模型的单位利润估计值从 0.088 可忽略地提高到 0.092。对于相同截止值（由于 0.5 是默认截止值），模型之间的性能差异为 0.004，这是由于 10x10

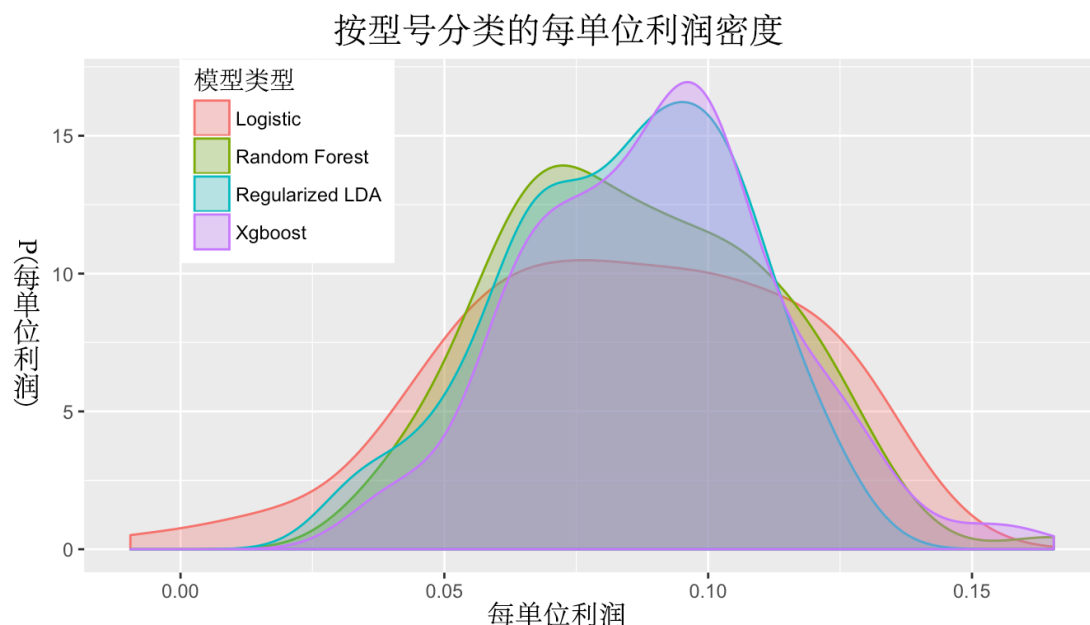
K-Fold CV 中用于最佳截止搜索过程的不同折叠集导致的差异。该结果证明了 CV 程序的有效性，因为两个完全不同的 CV 折叠组的性能估计具有相似的每单位利润值的中值。

## 结论：最终模型选择和指标

通过为模型选择最佳截止值，所有四个最终模型候选者都具有相似的性能指标，xgboost 和正则化 LDA 模型显示最佳单位利润估计值，逻辑回归和随机森林模型显示略差的结果。但是，单位利润估算范围仅为 0.005。因此，这些模型具有相似的性能特征。

为了进行最终选择，将具有最佳截止值的每个模型的 10 个重复 CV 折叠中的每一个的每单位利润估计的分布一起绘制在密度图中。密度图显示了 10x10 K-Fold CV 期间发生的值的分布。由于此程序产生的性能值是模型性能的估计值，因此可以将每个模型的每单位利润估计值的概率密度视为实际模型性能的抽样分布的代表。

```
1 # 合并上一节中的四个结果数据帧
2 all_perf = bind_rows(log_perf, rf_perf, xgb_perf, rlda_perf)
3 # 绘制分布图
4 all_perf %>% ggplot(aes(x = value, fill=group, color=group)) +
5   geom_density(alpha=0.3) +
6   theme(legend.position=c(0.2, 0.8), plot.title = element_text(hjust = 0.5)) +
7   labs(title = "按型号分类的每单位利润密度",
8        y = "P(每单位利润)",
9        x = "每单位利润",
10       fill = "模型类型",
11       color = "模型类型")
```



从这个图中可以看出，由于分布的高度重叠，没有一个模型明显优于另一个模型。但是，如果必须选择模型，xgboost 模型具有相对最有利的模型，其密度大于每单位利润估计值。相对较差的模型将是 Logistic 回归，因为它比其他模型具有更大的值变化，因此，当用于样本外数据时，预计会有更难以预测的性能。

最终模型适用于使用 10x10 K-Fold CV 的数据。为了在新预测上使用该模型，必须使用 0.7 的截止阈值来执行使用模型的“好”类的预测概率的条件变换。

```
1 set.seed(123)
2 # 创建自定义的indices: myFolds
3 myFolds <- createCVFolds(credit$creditability, k = 10)
4
5 # 创建可复用的trainControl object: myControl
6 myControl <- trainControl(summaryFunction = p35,
7                             classProbs = TRUE, # 重要!
8                             verboseIter = FALSE,
9                             savePredictions = TRUE,
10                             index = myFolds
11                             )
12
13 final_fit = train(creditability ~ .,
14                   data = credit,
15                   method = "xgbTree",
16                   tuneGrid = round1_tuneGridXGB,
17                   trControl = myControl,
18                   metric = "P35")
```

## 第四部分：确定最相关的变量

虽然模型选择过程已经完成，但我认为向管理者报告她提供的数据中的哪些变量高于平均信息以预测信誉是有用的。除了拥有自己的模型之外，她还可以利用这些信息为收集未来数据做出更好的决策，特别是在将相关申请人信息集中到高质量变量或开发新变量以获得更多细节方面。确定可信度的最强预测质量。

首先，数据集中的每个分类变量被分成称为虚拟变量的单独的二进制预测变量。

```
1 # 数据集中的每个分类变量被分成称为虚拟变量的单独的二进制预测变量
2 dummies <- dummyVars(creditability ~ ., data = credit)
3 dummies = predict(dummies, newdata = credit) %>% tbl_df
4 dummies$creditability = credit$creditability
```

然后，组合确定变量预测信息的两种方法，以确定每个变量的预测质量：



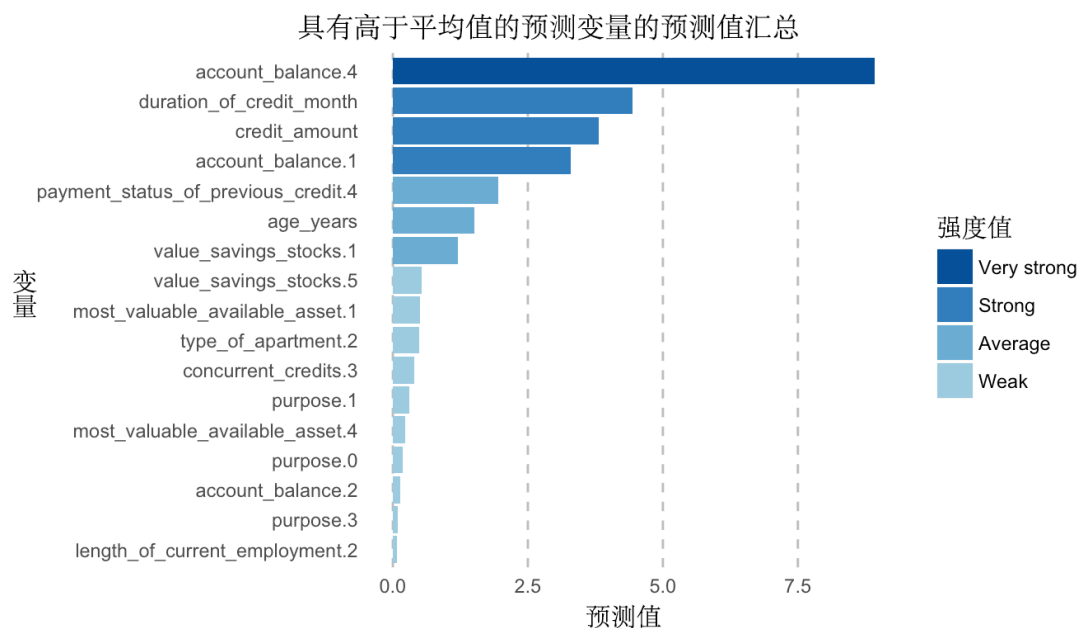
从 woe 包中，信息价值 (IV) 是风险管理中使用的概念，用于评估根据 WoE (证据权重) 计算的变量的预测能力。

从 xgboost 包中，该 xgb.importance 函数计算每个预测变量的增益。增益可以指示预测器在如何导致决策树分支 (例如 “xgbTree” 方法生成的分支) 方面的重要性。

IV 和增益都按比例缩放并标准化，然后进行平均。这些方法中的任何一种本身都是足够的，但是当组合在一起时该过程可以相互确认以增加对结果的信心。

```
1 # 确定最相关的变量
2 set.seed(123)
3 # 查找每个变量的信息值
4 invisible(capture.output(IV <- iv.mult(as.data.frame(dummies),"creditability", summary = TRUE)))
5
6 # 查找每个变量的收益
7 X = dummies %>% dplyr::select(-creditability)
8 library(xgboost)
9 # 为xgboost准备数据
10 x_num = as.matrix(X) %>% apply(2,as.numeric)
11 x_label = as.numeric(as.character(ifelse(dummies$creditability == "Good", 1, 0)))
12 x_matrix = xgboost::xgb.DMatrix(data = x_num, label = x_label)
13 # 使用分析中的xgboost设置拟合模型
14 bst <- xgboost(data = x_matrix,
15               nround = 300,
16               eta = 0.07,
17               max.depth = 2,
18               gamma = 0.1,
19               min_child_weight = 2,
20               subsample = 1,
21               colsample_bytree = 1,
22               objective = "binary:logistic",
23               verbose = 0)
24
25 # 确定最重要的预测因子
26 xgb_import = xgb.importance(colnames(x_num), do.NULL = TRUE, prefix = "col", model = bst)
27
28 # 结合两种类型的预测器质量测量的输出
29 combo = IV %>% left_join(xgb_import, by = c("Variable" = "Feature"))
30 combo[is.na(combo)] = 0
31
32 # 规范化并结合两种措施
33 combo.out = combo %>% dplyr::select(Variable, Strength, InformationValue, Gain) %>%
34   mutate(InformationValue = as.numeric(scale(InformationValue)), Gain = as.numeric(scale(Gain)),
35          Avg_Predict_Qual = InformationValue + Gain / 2) %>% arrange(Variable)
36
37 # 绘制结果
38 library(RColorBrewer)
39 fills <- rev(brewer.pal(6, "Blues"))
40 Variable <- InformationValue <- Strength <- NULL
41 ggplot(data = combo.out %>% filter(Avg_Predict_Qual > 0)) + geom_bar(aes(x = reorder(Variable,
42   Avg_Predict_Qual),
43   y = Avg_Predict_Qual, fill = Strength), stat = "identity") +
44   coord_flip() + scale_fill_manual(values = fills) + theme(panel.grid.major.y = element_blank(),
45   panel.grid.major.x = element_line(linetype = "dashed",
46   colour = "grey"), panel.grid.minor = element_blank(),
47   panel.background = element_blank(), axis.ticks.x = element_blank(),
48   axis.ticks.y = element_blank()) + xlab("变量") +
49   ylab("预测值") + ggtitle("具有高于平均值的预测变量的预测值汇总")
```





上面的图是管理者的潜在可交付成果之一，可以很好地总结结果并比较预测强度的差异。该 combo. out 表也可以共享，以提供更详细的结果。

似乎第四类 account\_balance 对确定可信度具有最高的预测价值。值得注意的是，不在此列表中的变量对可信度没有很强的预测价值。

## 第五部分：最终结论

xgboost 有时候特征重要性分析比随机森林还要准确，可见其强大之处，预测可信度的最佳模型使用 xgboost 算法将梯度提升树分类模型拟合到数据。

当使用 0.7 截止阈值时，最终模型预计产生每单位利润 0.092。与盲目比例模型相比，预期的每单位利润预期增长 0.131，该模型预测“良好”和“70-30 分割的信用可信度，预期每单位利润为-0.039”。如果模型符合此绩效评估，如果平均贷款金额为 10,000 美元，银行应该预计每个申请人的利润为 920 美元。