Central University of Finance and Economics

中央财经大学

现代统计软件课程

代码文档

吴宇翀

高思琴

陈蔚

指导老师：杨玥含

2020 年 6 月 20 日

```
knitr::opts_chunk$set(fig.pos = 'H', warning = FALSE, message = FALSE)
library(ggplot2)
library(caret)
library(kernlab)
library(pROC)
library(knitr)
library(magrittr)
# base_family = 'STXihei'
# bibliography: cite.bib
```

1 数据集说明

```
dat = read.csv("data.csv")
dat = dat[,-1]
dat$SeriousDlqin2yrs = as.factor(dat$SeriousDlqin2yrs)
```

```
explain = read.csv("dictionary_chinese.csv", header = TRUE)
kable(explain, caption = " 变量描述解释")
```

表 1: 变量描述解释

变量名	描述	变量类型
是否逾期	是否有超过 90 天的逾期	Y/N
无担保放款的循环利用	无分期付款债务的信用卡和个人信用额度总额	百分比
年龄	借款人年龄	整数
过去 2 年间逾期 30-59 天的次数	有逾期 30-59 天，但在过去 2 年没有更糟的情况出现的次数	整数
负债比率	每月债务支付，赡养费，生活费用除以月总收入	百分比
月收入	每月的收入	实数
未偿还贷款数量	开放式贷款的数量和信用额度（如信用卡）	整数
90 天逾期次数	借款人逾期 90 天或以上的次数	整数
不动产贷款或额度数量	按揭及房地产贷款数目，包括房屋净值信贷额度。	整数
过去 2 年逾期 60-89 天的次数	借款人逾期 60-89 天的次数，但过去两年更糟的情况出现	整数
家属人数	不包括自己在内的家属（配偶，子女等）数量。	整数

2 数据预处理

```
dat$RevolvingUtilizationOfUnsecuredLines[which(dat$RevolvingUtilizationOfUnsecuredLines < 0)] = 0
dat$RevolvingUtilizationOfUnsecuredLines[which(dat$RevolvingUtilizationOfUnsecuredLines > 1)] = 1
dat$DebtRatio[which(dat$DebtRatio < 0)] = 0
dat$DebtRatio[which(dat$DebtRatio > 1)] = 1
dat_complete = dat[complete.cases(dat),]
```

3 描述分析

3.1 年龄

```
ggplot(dat_complete, aes(x = age, fill = SeriousDlqin2yrs)) +
  geom_density(alpha = 0.3) +
  theme_minimal() +
  scale_fill_manual(values = c("#037418", "darkred"))
```

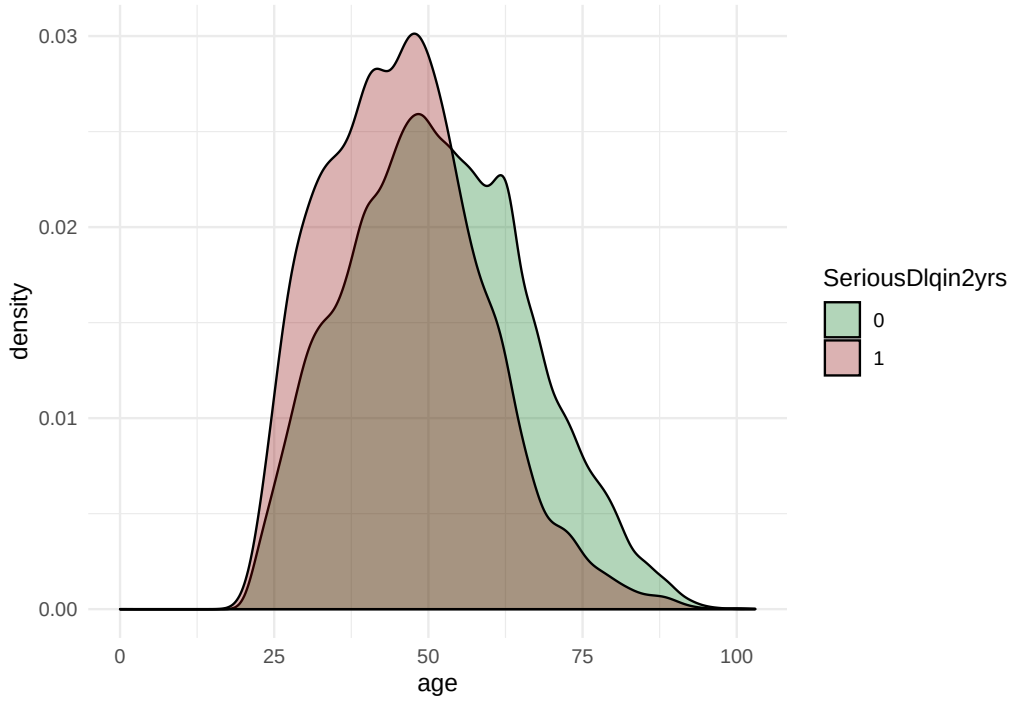

图 1: 信用卡逾期与否两类人群的年龄分布（红色代表逾期）

3.2 债务数量

```
dat_process = dat_complete[which(dat_complete$NumberRealEstateLoansOrLines < 5),]
good = dat_process[which(dat_process$SeriousDlqin2yrs == 0),]
bad = dat_process[which(dat_process$SeriousDlqin2yrs == 1),]
dat_process = rbind(good[1:1000,], bad[1:1000,])
ggplot(dat_process, aes(x = NumberRealEstateLoansOrLines, fill = SeriousDlqin2yrs)) +
  geom_histogram(stat = "count", alpha = 0.6) +
  theme_minimal() +
  scale_fill_manual(values = c("#037418", "darkred")) +
  facet_grid(cols = vars(SeriousDlqin2yrs)) +
  labs(y = "percentage")
```

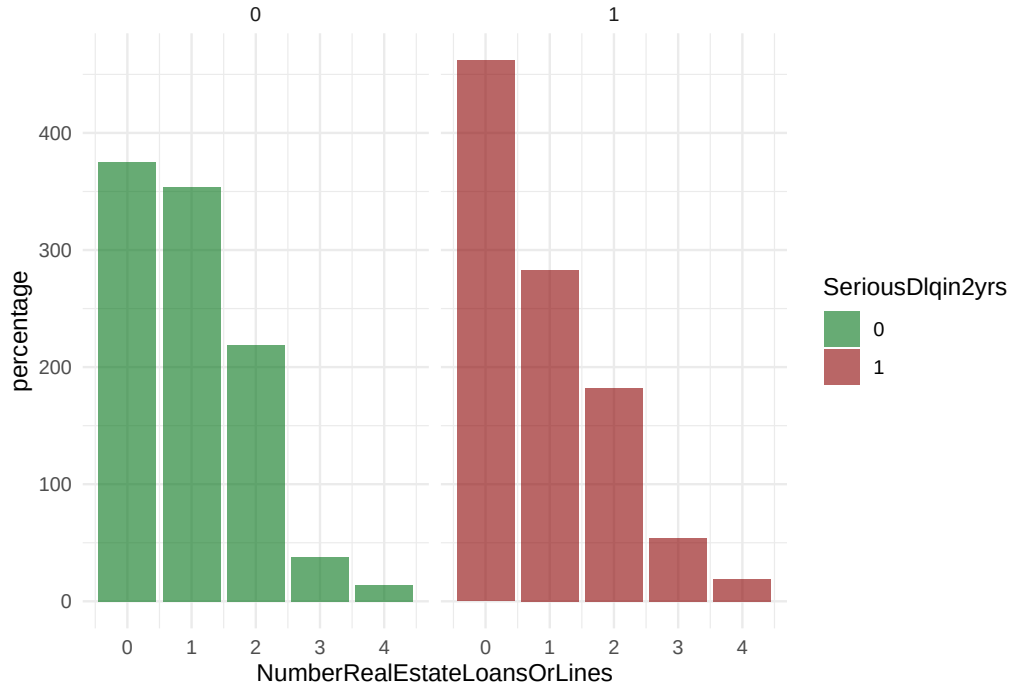

图 2: 信用卡逾期与否两类人群的债务数量（红色代表逾期）

3.3 月收入

```
dat_process = dat_complete[which(dat_complete$MonthlyIncome < 30000),]
ggplot(dat_process, aes(x = SeriousDlqin2yrs, y = MonthlyIncome, fill = SeriousDlqin2yrs)) +
  geom_violin(alpha = 0.3) +
  theme_minimal() +
  scale_fill_manual(values = c("#037418", "darkred"))
```

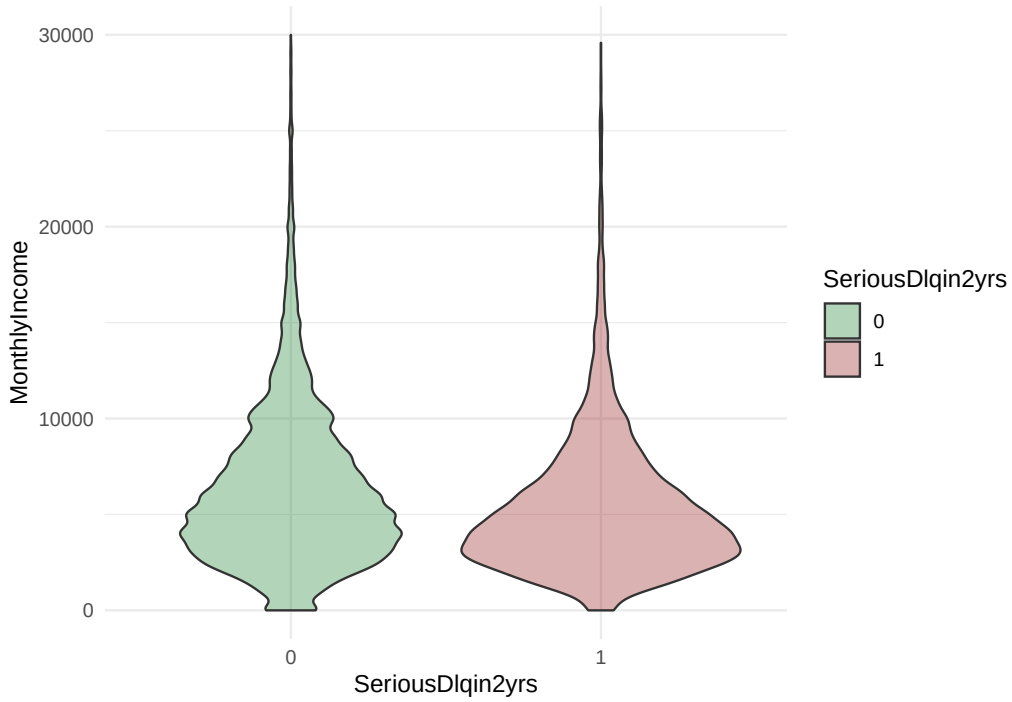

图 3: 信用卡逾期与否两类人群的月收入（红色代表逾期）

4 Logit 回归

4.1 拟合

```
set.seed(1)
inTraining <- createDataPartition(dat_complete$SeriousDlqin2yrs, p = .75, list = FALSE)
train <- dat_complete[inTraining,]
test <- dat_complete[-inTraining,]
```

```
logit2 = glm(SeriousDlqin2yrs ~ ., data = train, family = binomial(link = "logit"))
logit2_sum = summary(logit2)
translate = as.character(explain$变量名)
translate[1] = "（截距）"
rownames(logit2_sum$coefficients) = translate
kable(logit2_sum$coefficients, caption = "Logit 回归系数表", digit = 2)
```

表 2: Logit 回归系数表

	Estimate	Std. Error	z value	Pr(>|z|)
（截距）	-3.56	0.07	-51.33	0

	Estimate	Std. Error	z value	Pr(>|z|)
无担保放款的循环利用	2.47	0.04	57.73	0
年龄	-0.01	0.00	-11.96	0
过去 2 年间逾期 30-59 天的次数	0.32	0.01	22.80	0
负债比率	0.25	0.06	3.96	0
月收入	0.00	0.00	-7.31	0
未偿还贷款数量	0.03	0.00	8.73	0
90 天逾期次数	0.28	0.02	15.66	0
不动产贷款或额度数量	0.06	0.01	4.18	0
过去 2 年逾期 60-89 天的次数	-0.57	0.02	-26.53	0
家属人数	0.07	0.01	6.45	0

4.2 预测

```
probability = predict(logit2, test, type = "response")
distribution = as.data.frame(probability)
distribution = cbind(distribution, group = test$SeriousDlqin2yrs)
ggplot(distribution, aes(x = probability, fill = group)) +
  geom_density(alpha = 0.3) +
  theme_minimal() +
  scale_fill_manual(values = c("#037418", "darkred"))
```

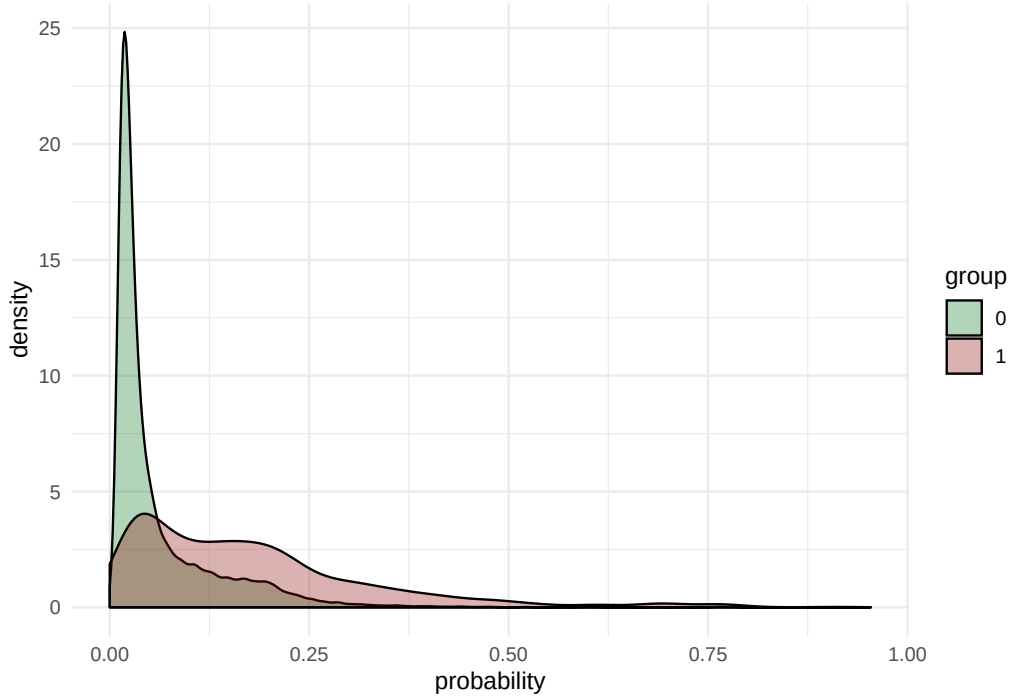

图 4: 预测的逾期概率值（红色代表已知为逾期）

```
testPred = probability
testPred[testPred > 0.5] = 1
testPred[testPred <= 0.5] = 0
testPred = as.factor(testPred)
```

4.3 混淆矩阵与验证结果

```
confusion = confusionMatrix(data = test$SeriousDlqin2yrs,
                reference = testPred,
                positive = "1")
kable(as.data.frame(confusion$table), caption = " 混淆矩阵表")
```

表 3: 混淆矩阵表

Prediction	Reference	Freq
0	0	27910
1	0	2003
0	1	68
1	1	86

```
table = as.data.frame(confusion$overall)
names(table) = c(" 指标值")
table = t(table)
rownames(table) = NULL
kable(table, caption = " 验证结果表", digit = 3)
```

表 4: 验证结果表

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull	AccuracyPValue	McnemarPValue
0.931	0.068	0.928	0.934	0.995	1	0

```
table = as.data.frame(confusion$byClass[1:5])
names(table) = c(" 指标值")
table = t(table)
kable(table, caption = " 灵敏度和特异度等指标表", digit = 3)
```

表 5: 灵敏度和特异度等指标表

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision
指标值	0.558	0.933	0.041	0.998	0.041

4.4 接受者操作特征（ROC）曲线

```
rocCurve = roc(response = test$SeriousDlqin2yrs,
               predictor = probability,
               levels = rev(levels(test$SeriousDlqin2yrs)),
               plot = TRUE,
               print.thres=TRUE, print.auc=TRUE)
```

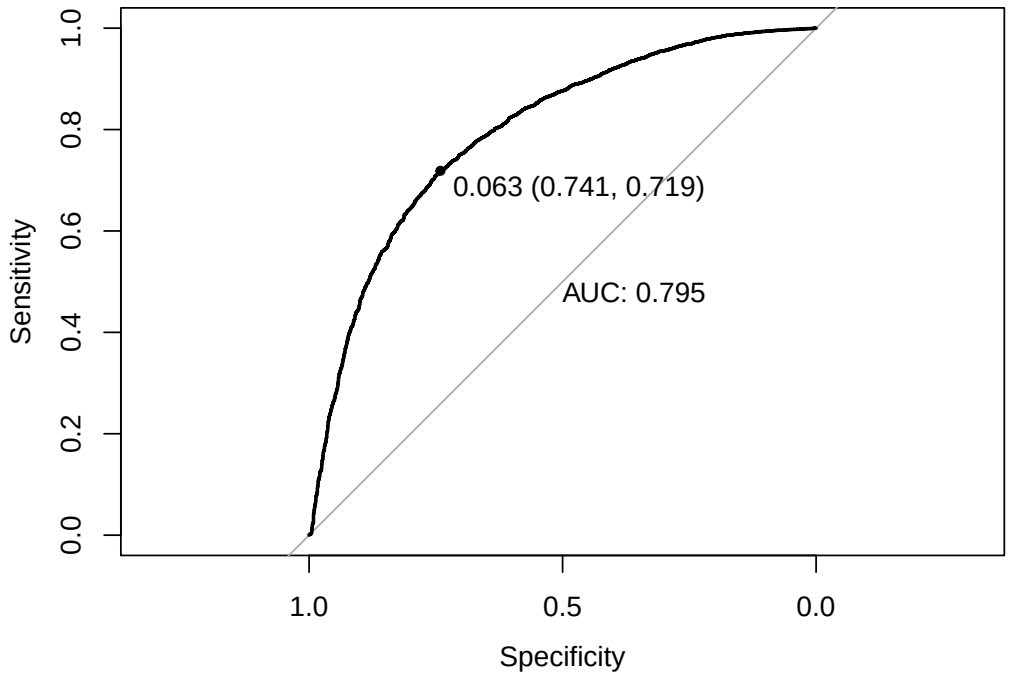

图 5: Logit 模型的 ROC 曲线

5 模型选择

5.1 抽样、训练与评价指标

```
set.seed(1)
inTraining <- createDataPartition(dat_complete$SeriousDlqin2yrs, p = .01, list = FALSE)
training <- dat_complete[inTraining,]

fitControl <- trainControl(## 10-fold CV
                           method = "repeatedcv",
                           number = 10,
                           ## repeated ten times
                           repeats = 5)
```

5.2 Logit 回归

```
set.seed(1)
logit <- train(SeriousDlqin2yrs ~ ., data = training,
```

```
                  method = "glm",
                  trControl = fitControl)
table = logit$results
rownames(table) = NULL
kable(table, caption = " 在重抽样下 Logit 模型的表现", digits = 3)
```

表 6: 在重抽样下 Logit 模型的表现

parameter	Accuracy	Kappa	AccuracySD	KappaSD
none	0.931	0.202	0.014	0.173

5.3 线性判别分析（LDA）

```
set.seed(1)
lda <- train(SeriousDlqin2yrs ~ ., data = training,
                  method = "lda",
                  trControl = fitControl,
                preProc = c("center", "scale"))
table = lda$results
rownames(table) = NULL
kable(table, caption = " 在重抽样下 LDA 模型的表现", digits = 3)
```

表 7: 在重抽样下 LDA 模型的表现

parameter	Accuracy	Kappa	AccuracySD	KappaSD
none	0.925	0.122	0.013	0.161

```
trellis.par.set(caretTheme())
densityplot(lda, pch = "|")
```

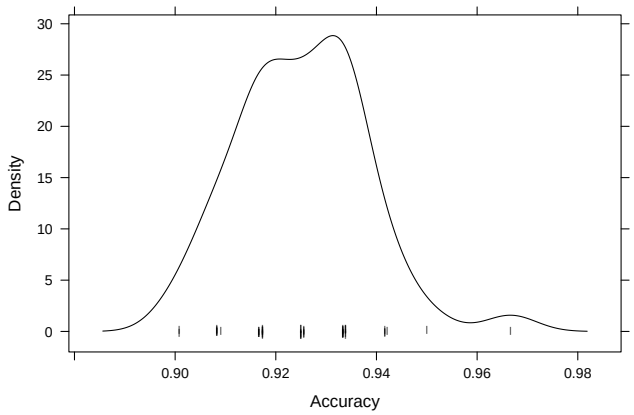

图 6: 在重抽样下 LDA 模型的准确率分布

5.4 偏最小二乘判别分析（PLSDA）

```
set.seed(1)
plsda <- train(SeriousDlqin2yrs ~ ., data = training,
                method = "pls",
                trControl = fitControl,
              tuneGrid = expand.grid(.ncomp = 1:10))
table = plsda$results
rownames(table) = NULL
kable(table, caption = " 在重抽样下 PLSDA 模型的表现", digits = 3)
```

表 8: 在重抽样下 PLSDA 模型的表现

ncomp	Accuracy	Kappa	AccuracySD	KappaSD
1	0.930	0.000	0.004	0.000
2	0.930	0.000	0.004	0.000
3	0.931	0.021	0.005	0.062
4	0.930	0.018	0.006	0.056
5	0.930	0.025	0.007	0.084
6	0.930	0.024	0.008	0.086
7	0.930	0.024	0.008	0.086
8	0.930	0.024	0.007	0.085
9	0.930	0.024	0.007	0.085
10	0.930	0.024	0.007	0.085

```r
trellis.par.set(caretTheme())
plot(plsda, metric = "Kappa")
plot(plsda)
```

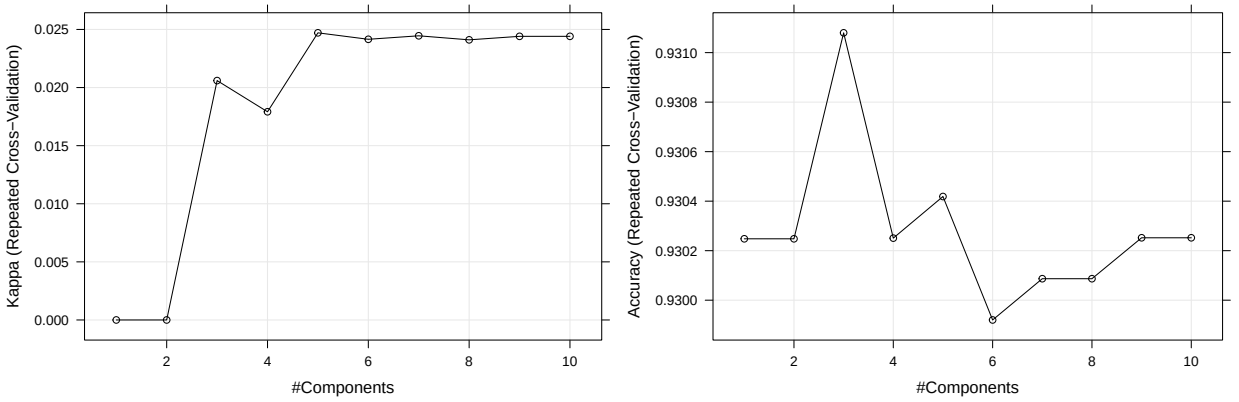

图 7: Kappa 和准确率指标随主成分个数的变化

```r
plsImp = varImp(plsda, scale = FALSE)
table = data.frame(variables = rownames(plsImp$importance), importence = plsImp$importance$Overall
ggplot(table, aes(x = reorder(variables, importence), y = importence)) +
  geom_col() +
  theme_minimal() +
  coord_flip() +
  labs(x = "variables")
```

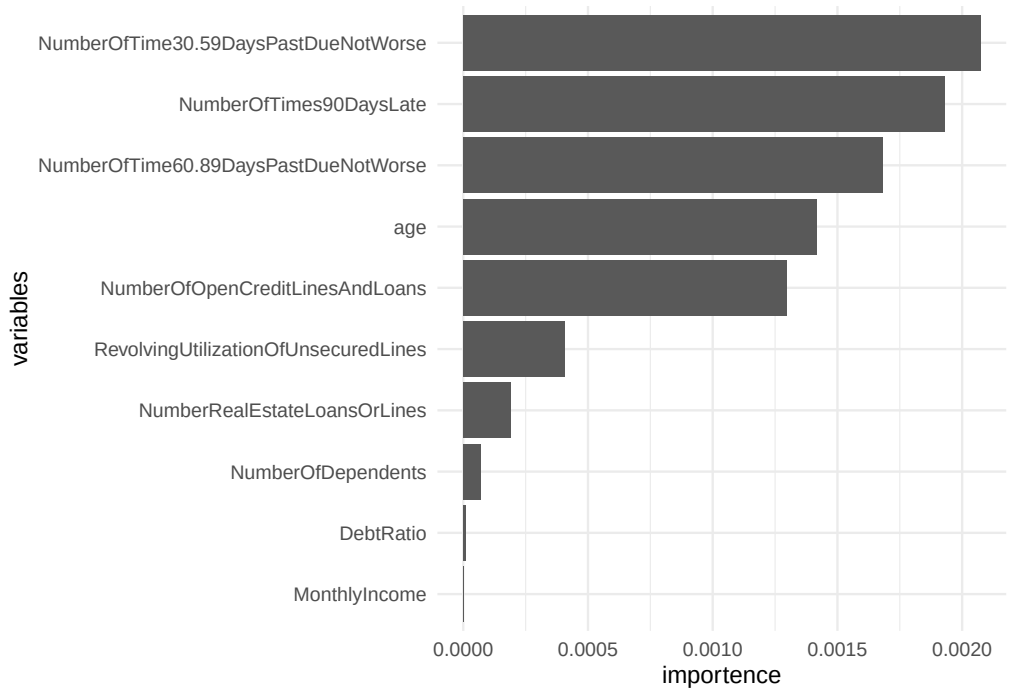

图 8: 变量重要程度

5.5 SVM

```
set.seed(1)
svm <- train(SeriousDlqin2yrs ~ ., data = training,
                 method = "svmRadial",
                 trControl = fitControl,
             tuneLength = 5)
kable(svm$results, caption = " 在重抽样下 SVM 模型的表现", digits = 3)
```

表 9: 在重抽样下 SVM 模型的表现

sigma	C	Accuracy	Kappa	AccuracySD	KappaSD
0.149	0.25	0.930	0.000	0.004	0.000
0.149	0.50	0.930	0.000	0.004	0.000
0.149	1.00	0.929	-0.003	0.005	0.006
0.149	2.00	0.928	0.051	0.008	0.095
0.149	4.00	0.928	0.142	0.012	0.147

```
trellis.par.set(caretTheme())
plot(svm)
plot(svm, metric = "Kappa")
```

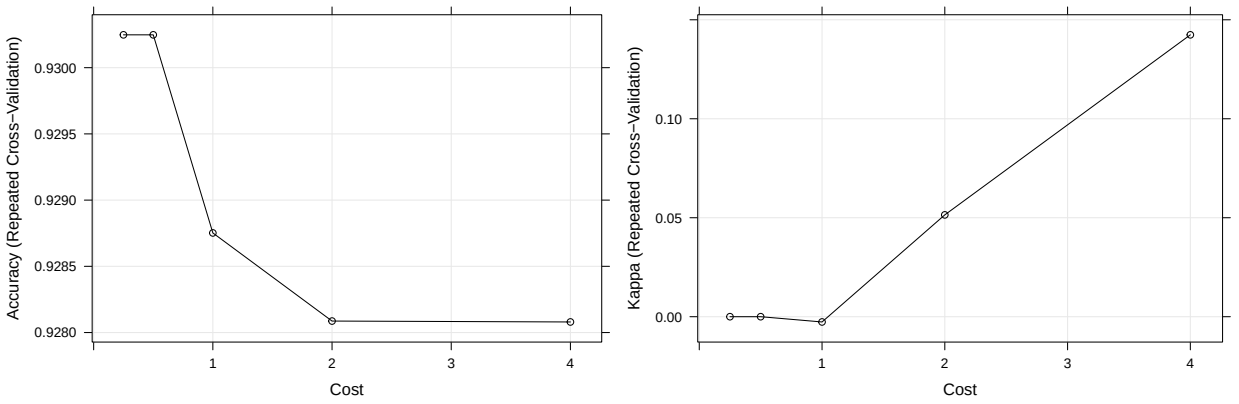

图 9: 调优参数不同取值下的准确率和 Kappa 指标变化

5.6 随机梯度助推法（GBM）

```
trellis.par.set(caretTheme())
plot(gbm)

trellis.par.set(caretTheme())
plot(gbm, metric = "Kappa")
```

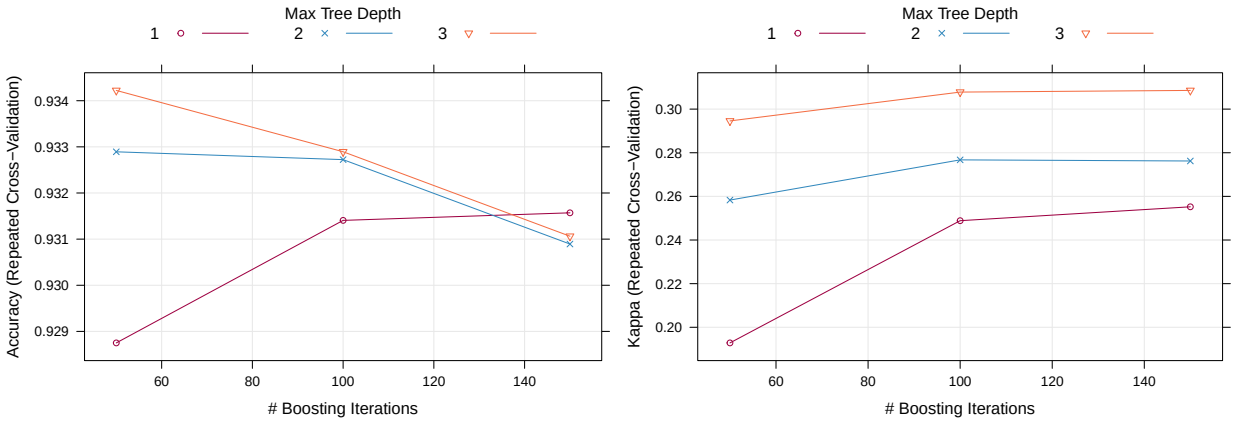

图 10: 调优参数和迭代次数不同取值下的准确率和 Kappa 指标变化

```
trellis.par.set(caretTheme())
densityplot(gbm, pch = "|")
```

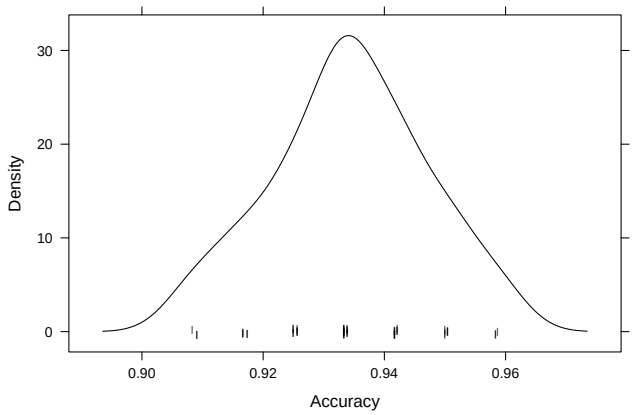

图 11: 在重抽样下 GBM 模型的准确率分布

5.7 模型间的比较

```
resamp = resamples(list(LDA = lda, PLSDA = plsda, SVM = svm, GBM = gbm, Logit = logit))
s1 = summary(resamp)
s2 = summary(diff(resamp))
```

```
ggplot(resamp,
       models = c("LDA", "PLSDA", "GBM", "Logit"),
       metric = "Kappa",
       conf.level = 0.95) +
  theme_bw()
```

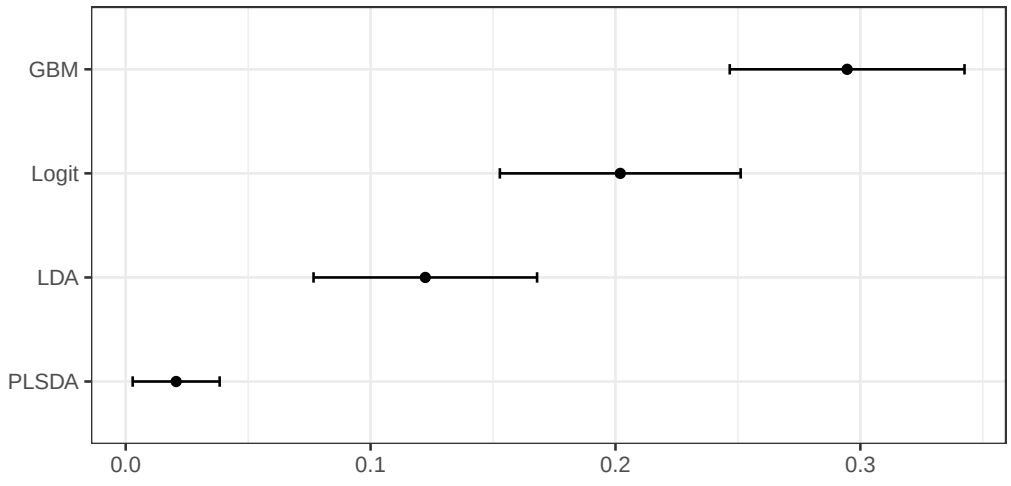

图 12: 模型间 Kappa 的比较（0.95 置信区间）

```
ggplot(resamp,
       models = c("LDA", "PLSDA", "SVM", "GBM", "Logit"),
       metric = "Accuracy",
       conf.level = 0.95) +
  theme_bw()
```

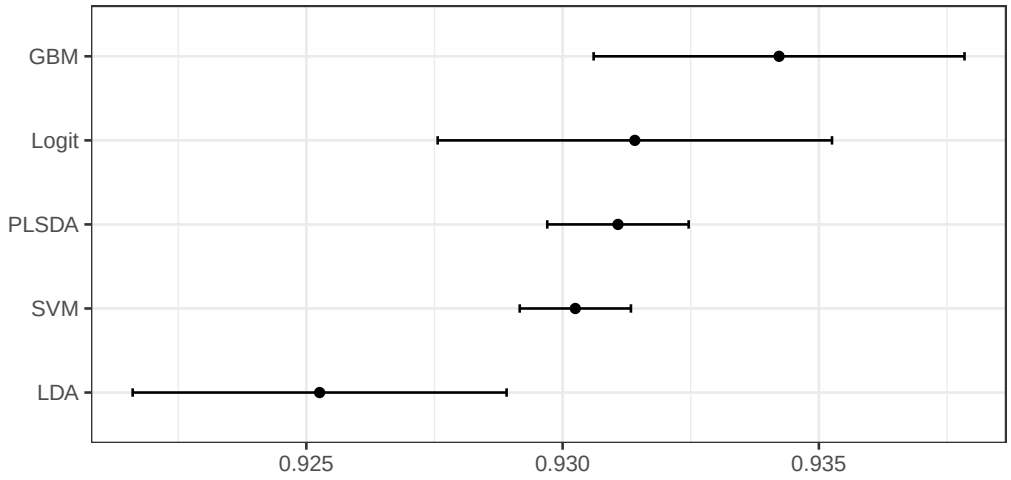

图 13: 模型间准确率的比较（0.95 置信区间）

6 附录

6.1 模型间准确率和 Kappa 的比较

```
kable(s1$statistics$Accuracy, caption = " 模型间准确率的比较", digit = 3)
```

表 10: 模型间准确率的比较

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
LDA	0.901	0.917	0.925	0.925	0.933	0.967	0
PLSDA	0.926	0.926	0.933	0.931	0.933	0.942	0
SVM	0.926	0.926	0.933	0.930	0.933	0.933	0
GBM	0.908	0.926	0.933	0.934	0.942	0.959	0
Logit	0.901	0.925	0.933	0.931	0.942	0.975	0

```
kable(s2$table$Accuracy, caption = " 模型间准确率差异矩阵", digit = 3)
```

表 11: 模型间准确率差异矩阵

	LDA	PLSDA	SVM	GBM	Logit
LDA		-0.0058209	-0.0049890	-0.0089642	-0.0061501
PLSDA	0.027190		0.0008320	-0.0031433	-0.0003292
SVM	0.076479	0.237793		-0.0039752	-0.0011612
GBM	0.001116	0.929403	0.356758		0.0028140
Logit	0.003293	1.000000	1.000000	1.000000	

```
kable(s1$statistics$Kappa, caption = " 模型间 Kappa 的比较", digit = 3)
```

表 12: 模型间 Kappa 的比较

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
LDA	-0.039	-0.015	0.118	0.122	0.188	0.732	0
PLSDA	0.000	0.000	0.000	0.021	0.000	0.211	0
SVM	0.000	0.000	0.000	0.000	0.000	0.000	0
GBM	-0.038	0.181	0.302	0.295	0.422	0.597	0
Logit	-0.038	0.105	0.183	0.202	0.322	0.757	0

```
kable(s2$table$Accuracy, caption = " 模型间 Kappa 差异矩阵", digit = 3)
```

表 13: 模型间 Kappa 差异矩阵

	LDA	PLSDA	SVM	GBM	Logit
LDA		-0.0058209	-0.0049890	-0.0089642	-0.0061501
PLSDA	0.027190		0.0008320	-0.0031433	-0.0003292
SVM	0.076479	0.237793		-0.0039752	-0.0011612
GBM	0.001116	0.929403	0.356758		0.0028140
Logit	0.003293	1.000000	1.000000	1.000000	

6.2 Logit 回归结果

```
logit2_sum
```

```
##
## Call:
```

```
## glm(formula = SeriousDlqin2yrs ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5488  -0.3724  -0.2387  -0.1852   4.5549
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## （截距）                     -3.559e+00  6.933e-02 -51.331  < 2e-16 ***
## 无担保放款的循环利用          2.471e+00  4.280e-02  57.727  < 2e-16 ***
## 年龄                         -1.368e-02  1.143e-03 -11.962  < 2e-16 ***
## 过去2年间逾期30-59天的次数    3.177e-01  1.393e-02  22.801  < 2e-16 ***
## 负债比率                      2.466e-01  6.234e-02   3.956 7.63e-05 ***
## 月收入                       -2.978e-05  4.071e-06  -7.314 2.59e-13 ***
## 未偿还贷款数量                2.842e-02  3.255e-03   8.730  < 2e-16 ***
## 90天逾期次数                  2.818e-01  1.800e-02  15.660  < 2e-16 ***
## 不动产贷款或额度数量          5.884e-02  1.407e-02   4.182 2.89e-05 ***
## 过去2年逾期60-89天的次数     -5.721e-01  2.157e-02 -26.526  < 2e-16 ***
## 家属人数                      7.441e-02  1.154e-02   6.451 1.11e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 45518  on 90201  degrees of freedom
## Residual deviance: 38190  on 90191  degrees of freedom
## AIC: 38212
##
## Number of Fisher Scoring iterations: 6
```

6.3 数据

```
str(dat)
```

```
## 'data.frame':    150000 obs. of  11 variables:
##  $ SeriousDlqin2yrs                    : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ RevolvingUtilizationOfUnsecuredLines: num  0.766 0.957 0.658 0.234 0.907 ...
##  $ age                                 : int  45 40 38 30 49 74 57 39 27 57 ...
```

```
##  $ NumberOfTime30.59DaysPastDueNotWorse: int  2 0 1 0 1 0 0 0 0 0 ...
##  $ DebtRatio                           : num  0.803 0.1219 0.0851 0.036 0.0249 ...
##  $ MonthlyIncome                       : int  9120 2600 3042 3300 63588 3500 NA 3500 NA 23684 .
##  $ NumberOfOpenCreditLinesAndLoans     : int  13 4 2 5 7 3 8 8 2 9 ...
##  $ NumberOfTimes90DaysLate             : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ NumberRealEstateLoansOrLines        : int  6 0 0 0 1 1 3 0 0 4 ...
##  $ NumberOfTime60.89DaysPastDueNotWorse: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NumberOfDependents                  : int  2 1 0 0 0 1 0 0 NA 2 ...
```

```
summary(dat)
```

```
##  SeriousDlqin2yrs RevolvingUtilizationOfUnsecuredLines      age
##  0:139974         Min.   :0.00000                     Min.   :  0.0
##  1: 10026         1st Qu.:0.02987                     1st Qu.: 41.0
##                   Median :0.15418                     Median : 52.0
##                   Mean   :0.31920                     Mean   : 52.3
##                   3rd Qu.:0.55905                     3rd Qu.: 63.0
##                   Max.   :1.00000                     Max.   :109.0
##
##  NumberOfTime30.59DaysPastDueNotWorse   DebtRatio      MonthlyIncome
##  Min.   : 0.000                       Min.   :0.0000   Min.   :      0
##  1st Qu.: 0.000                       1st Qu.:0.1751   1st Qu.:   3400
##  Median : 0.000                       Median :0.3665   Median :   5400
##  Mean   : 0.421                       Mean   :0.4663   Mean   :   6670
##  3rd Qu.: 0.000                       3rd Qu.:0.8683   3rd Qu.:   8249
##  Max.   :98.000                       Max.   :1.0000   Max.   :3008750
##                                                        NA's   :29731
##  NumberOfOpenCreditLinesAndLoans NumberOfTimes90DaysLate
##  Min.   : 0.000                  Min.   : 0.000
##  1st Qu.: 5.000                  1st Qu.: 0.000
##  Median : 8.000                  Median : 0.000
##  Mean   : 8.453                  Mean   : 0.266
##  3rd Qu.:11.000                  3rd Qu.: 0.000
##  Max.   :58.000                  Max.   :98.000
##
##  NumberRealEstateLoansOrLines NumberOfTime60.89DaysPastDueNotWorse
##  Min.   : 0.000               Min.   : 0.0000
##  1st Qu.: 0.000               1st Qu.: 0.0000
##  Median : 1.000               Median : 0.0000
##  Mean   : 1.018               Mean   : 0.2404
```

```
##  3rd Qu.: 2.000              3rd Qu.: 0.0000
##  Max.   :54.000              Max.   :98.0000
##
##  NumberOfDependents
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 0.757
##  3rd Qu.: 1.000
##  Max.   :20.000
##  NA's   :3924
```