Análise Preditiva

Mayara Yonemura - A58337141

05/09/2021

Briefing

Os dados disponibilizados contém informações socio-econômicas de alunos que contrataram um financiamento estudantil por meio do FIES.

Objetivo da Análise Preditiva

Utilizar um modelo de predição, aprendido em aula, para classificar os alunos em bons pagadores ou mau pagadores.

```
- Considere alto o custo de um falso negativo (pior classificar um aluno como bom pagador qua
ndo seria um mau pagador)
- Destacar sensibilidade do modelo
```

Resumo Executivo

Por se tratar de um problema de classificação, no qual precisa ser previsto se, a partir de determinadas informações fornecidas pelo aluno, o mesmo será inadimplente ou não, os modelos testados foram dois: Regressão Logística e Árvore de Decisão. Para Regressão Logística foram testadas algumas técnicas para conseguir otimizar o modelo, sendo elas: Padronização das informações, convertendo todas as variáveis númericas para a mesma escala, a Regularização LASSO, e a seleção de variáveis utilizando K-Fold Cross Validation e a própria saída da técnica LASSO que nos diz as correlações das variáveis através do menor coeficiente.

Inicialmente foi utilizado o Dataset inteiro, com todas as variáveis, utilizando o modelo de Regressão Logística com Padronização, transformando todas as variáveis categóricas em fatores, e todos as variáveis númericas na mesma escala, e também a técnica de Regularização LASSO, o qual obteve o MSE (erro médio) melhor quanto comparado ao modelo padronizado. Assim, ainda com a técnica LASSO foi feita a seleção de variáveis e testando o modelo novamente, o qual teve aumento no MSE, ou seja, o modelo se adaptou menos (errou mais).

Em sequência foi testado a mesma seleção de variáveis com o Dataset padronizado, removendo os Outliers do Dataset, com 61% de Acurácia e 80% de Sensibilidade. Com o modelo treinado, foi testada a base de Outliers removida do Dataset original, o qual mostrou acurácia de 74%, aumentando a eficiência do modelo.

Por fim, também foi testado um modelo de Árvore de Decisão, porém com a eficiência do modelo menor que os dois de Regressão Logística, com a acurácia em 62% e a sensbilidade em 77%.

Visto que, aplicando as técnicas de Regularização LASSO com seleção de variáveis aumentou o erro médio padrão do modelo, o mesmo teria que ser implementado com a coleta de todas as informações do Dataset - são 13 variáveis ao todo -, assim sendo, eliminando o modelo de

Árvore de Decisão por resultados de acurácia (65% LASSO x 62% Árvore de Decisão) que também foi treinado com todas as variáveis. Considerando que a seleção de variáveis diminui a quantidade de informações requeridas, deixando o modelo menos complexo, o melhor modelo foi o de Regressão Logística com Padronização, selecionando 6 variáveis preditoras.

Melhor Modelo

Regressão Logística com Padronização e Feature Selection

```
summary(modeloreglog4)
```

```
##
## Call:
## glm(formula = ST_INADIMPLENCIA ~ VL_RENDA_FAMILIAR_BRUTA_MENSAL +
##     ST_ENSINO_MEDIO_ESCOLA_PUBLICA + DS_ESTADO_CIVIL + VL_FINANCIAMENTO +
##     SG_SEXO + ST_BOLSISTA_PROUNI, family = "binomial", data = dftreinocapita)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6964  -1.2188   0.8867   1.0261   2.8376
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     0.022730   0.023193   0.980 0.327069
## VL_RENDA_FAMILIAR_BRUTA_MENSAL -0.724870   0.010215 -70.963  < 2e-16 ***
## ST_ENSINO_MEDIO_ESCOLA_PUBLICAP 0.241540   0.025462   9.486  < 2e-16 ***
## ST_ENSINO_MEDIO_ESCOLA_PUBLICAS 0.341156   0.016112  21.174  < 2e-16 ***
## DS_ESTADO_CIVILDIVORCIADO      -0.232124   0.045725  -5.077 3.84e-07 ***
## DS_ESTADO_CIVILSEPARADO        -0.262000   0.068887  -3.803 0.000143 ***
## DS_ESTADO_CIVILSOLTEIRO        -0.228415   0.019119 -11.947  < 2e-16 ***
## DS_ESTADO_CIVILUNIAO ESTAVEL   -0.110261   0.040247  -2.740 0.006151 **
## DS_ESTADO_CIVILVIUVO           -0.186127   0.160807  -1.157 0.247086
## VL_FINANCIAMENTO               -0.115883   0.007683 -15.083  < 2e-16 ***
## SG_SEXOMasculino               -0.076883   0.012002  -6.406 1.49e-10 ***
## ST_BOLSISTA_PROUNIS            -0.518606   0.034029 -15.240  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 171176  on 124097  degrees of freedom
## Residual deviance: 162968  on 124086  degrees of freedom
## AIC: 162992
##
## Number of Fisher Scoring iterations: 4
```

```
confusionMatrix(table(data = previsoes4, reference = targettestecapita), positive = '1')
```

```
## Confusion Matrix and Statistics
##
##     reference
## data     0     1
##    0  9667  5770
##    1 14778 22971
##
##                Accuracy : 0.6137
##                  95% CI : (0.6095, 0.6178)
##     No Information Rate : 0.5404
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2002
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7992
##             Specificity : 0.3955
##          Pos Pred Value : 0.6085
##          Neg Pred Value : 0.6262
##              Prevalence : 0.5404
##          Detection Rate : 0.4319
##    Detection Prevalence : 0.7098
##       Balanced Accuracy : 0.5974
##
##        'Positive' Class : 1
##
```

Modelo Alternativo

Regressão Logística com Regularização LASSO

```
summary(modeloregloglasso)
```

```
##
## Call:
## glm(formula = as.formula("ST_INADIMPLENCIA ~ ."), family = binomial(link = "logit"),
##     data = dftreinolasso)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.174  -1.098   0.685   1.004   4.670
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     6.327e-01  9.215e-02   6.865 6.63e-12 ***
## SG_SEXOMasculino               -8.718e-02  1.212e-02  -7.195 6.23e-13 ***
## NU_IDADE                        8.418e-03  1.212e-03   6.944 3.81e-12 ***
## DS_RACA_CORBRANCO              -1.898e-01  3.648e-02  -5.202 1.97e-07 ***
## DS_RACA_CORINDIO                2.333e-01  1.231e-01   1.895 0.058138 .
## DS_RACA_CORNEGRO                4.382e-01  4.000e-02  10.955  < 2e-16 ***
## DS_RACA_CORPARDO                1.145e-01  3.638e-02   3.148 0.001643 **
## SG_UFAL                        -5.584e-02  8.207e-02  -0.680 0.496273
## SG_UFAM                         7.574e-01  1.008e-01   7.514 5.74e-14 ***
## SG_UFAP                         5.970e-01  1.007e-01   5.928 3.06e-09 ***
## SG_UFBA                        -3.096e-01  7.212e-02  -4.293 1.77e-05 ***
## SG_UFCE                        -3.298e-01  7.376e-02  -4.471 7.79e-06 ***
## SG_UFDF                        -1.284e-01  7.281e-02  -1.763 0.077904 .
## SG_UFES                        -6.211e-01  9.107e-02  -6.819 9.14e-12 ***
## SG_UFGO                        -1.444e-01  7.327e-02  -1.970 0.048813 *
## SG_UFMA                        -1.298e-01  7.698e-02  -1.686 0.091842 .
## SG_UFMG                        -5.736e-01  7.073e-02  -8.110 5.06e-16 ***
## SG_UFMS                        -1.606e-02  7.440e-02  -0.216 0.829140
## SG_UFMT                        -4.724e-02  7.236e-02  -0.653 0.513854
## SG_UFPA                         2.709e-01  8.407e-02   3.222 0.001271 **
## SG_UFPB                        -3.979e-01  8.009e-02  -4.968 6.75e-07 ***
## SG_UFPE                        -7.790e-02  7.484e-02  -1.041 0.297891
## SG_UFPI                        -9.270e-01  8.724e-02 -10.627  < 2e-16 ***
## SG_UFPR                        -7.796e-01  7.160e-02 -10.887  < 2e-16 ***
## SG_UFRJ                         3.319e-01  7.575e-02   4.381 1.18e-05 ***
## SG_UFRN                        -1.763e-01  7.602e-02  -2.319 0.020392 *
## SG_UFRO                        -8.376e-01  1.015e-01  -8.249  < 2e-16 ***
## SG_UFRR                        -7.262e-02  1.155e-01  -0.629 0.529506
## SG_UFRS                        -6.730e-01  7.334e-02  -9.177  < 2e-16 ***
## SG_UFSC                        -1.064e+00  7.497e-02 -14.191  < 2e-16 ***
## SG_UFSE                        -2.380e-01  8.443e-02  -2.819 0.004817 **
## SG_UFSP                        -2.367e-03  7.381e-02  -0.032 0.974418
## SG_UFTO                        -9.316e-01  9.002e-02 -10.349  < 2e-16 ***
## DS_ESTADO_CIVILDIVORCIADO      -2.500e-01  4.527e-02  -5.522 3.36e-08 ***
## DS_ESTADO_CIVILSEPARADO        -3.010e-01  6.981e-02  -4.311 1.62e-05 ***
## DS_ESTADO_CIVILSOLTEIRO        -1.118e-01  2.005e-02  -5.577 2.45e-08 ***
## DS_ESTADO_CIVILUNIAO  ESTAVEL   1.004e-01  3.881e-02   2.587 0.009678 **
## DS_ESTADO_CIVILVIUVO           -3.303e-01  1.582e-01  -2.088 0.036829 *
## ST_DEFICIENCIAS                -8.841e-02  8.172e-02  -1.082 0.279287
## ST_ENSINO_MEDIO_ESCOLA_PUBLICAP  2.830e-01  2.536e-02  11.162  < 2e-16 ***
## ST_ENSINO_MEDIO_ESCOLA_PUBLICAS  4.004e-01  1.676e-02  23.888  < 2e-16 ***
## ST_BOLSISTA_PROUNIS            -4.661e-01  3.396e-02 -13.724  < 2e-16 ***
## VL_RENDA_FAMILIAR_BRUTA_MENSAL -1.878e-04  5.420e-06 -34.656  < 2e-16 ***
## VL_RENDA_PESSOA_BRUTA_MENSAL   -1.835e-05  9.395e-06  -1.953 0.050792 .
## VL_RENDA_PERCAPITA             -3.098e-04  1.582e-05 -19.580  < 2e-16 ***
## NU_SEMESTRE_FINANCIADO          9.959e-03  2.972e-03   3.351 0.000804 ***
```

```
## VL_FINANCIAMENTO               -3.523e-06  2.566e-07 -13.731  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 184378 on 133136 degrees of freedom
## Residual deviance: 166214 on 133090 degrees of freedom
## AIC: 166308
##
## Number of Fisher Scoring iterations: 4
```

```
confusionMatrix(table(data = as.factor(previsoeslasso), reference = as.factor(dftestelasso$ST
_INADIMPLENCIA)), positive = '1')
```

```
## Confusion Matrix and Statistics
##
##     reference
## data     0     1
##    0 15447  8077
##    1 11829 21706
##
##                Accuracy : 0.6511
##                  95% CI : (0.6472, 0.655)
##     No Information Rate : 0.522
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2968
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7288
##             Specificity : 0.5663
##          Pos Pred Value : 0.6473
##          Neg Pred Value : 0.6566
##              Prevalence : 0.5220
##          Detection Rate : 0.3804
##    Detection Prevalence : 0.5877
##       Balanced Accuracy : 0.6476
##
##        'Positive' Class : 1
##
```

Códigos Utilizados

Carregando as bibliotecas

```
library(ggplot2)
library(ggcorrplot)
library(dplyr)
library(e1071)
library(caret)
library(ROCR)
library(pROC)
library(plyr)
library(knitr)
library(rmarkdown)
library(htmltools)
library(psych)
library(gmodels)
library(glmnet)
library(ridge)
library(rpart)
library(randomForest)
```

Ingestão dos Dados

```
setwd("")
dtable <- read.csv('dados_trab_final_AP.csv', header=TRUE, sep = ';', dec = ',', stringsAsFac
tors = FALSE)
```

Dimensões do Dataset

- Colunas e Linhas

```
dim(dtable)
```

Visualização das 5 primeiras linhas do Dataset

```
##View(dtable)
head(dtable)
```

Visualização das 5 últimas linhas do Dataset

```
tail(dtable)
```

Contagem de dados nulos por coluna:

```
colSums(is.na(dtable))
```

Visualização dos datatypes do Dataset

```
str(dtable)
```

Convertendo a tabela em Dataframe

```
dffinanciamento <- data.frame(dtable)
```

Explorando Variáveis Númericas

Resumo Estatístico

```
summary(dtable[c("NU_IDADE", "VL_RENDA_FAMILIAR_BRUTA_MENSAL", "VL_RENDA_PESSOA_BRUTA_MENSAL"
,
                 "VL_RENDA_PERCAPITA", "NU_SEMESTRE_FINANCIADO", "VL_FINANCIAMENTO")])
```

Distribuição de Frequência de Idade

```
hist(dffinanciamento$NU_IDADE,
                  main= "Distribuição de Frequência de Idade",
                  xlab = "Idade")
```

BoxPlot da Quantidade de Semestres Financiados

```
hist(dffinanciamento$NU_SEMESTRE_FINANCIADO,
     main = "Quantidade de Semestres Financiados",
     xlab = "Quantidade de Semestres")
```

BoxPlot dos valores de Renda

```
boxplot(dffinanciamento$VL_RENDA_FAMILIAR_BRUTA_MENSAL,  dffinanciamento$VL_RENDA_PESSOA_BRUTA
_MENSAL, dffinanciamento$VL_RENDA_PERCAPITA,
        main = "BoxPlot dos valores de Rendas",
        names = c("Renda Familia Bruta", "Renda Pessoa Bruta", "Renda Per Capita"),
        ylab = "Valor (R$)")
```

Painéis de Relacionamento entre as variáveis numéricas

```
pairs.panels(dffinanciamento[, c("NU_IDADE", "VL_RENDA_FAMILIAR_BRUTA_MENSAL", "VL_RENDA_PESS
OA_BRUTA_MENSAL",
                                 "VL_RENDA_PERCAPITA", "NU_SEMESTRE_FINANCIADO", "VL_FINANCIA
MENTO")])
```

Análise de Correlação entre as variáveis Númericas

```
dffinanum <- dffinanciamento[, c("NU_IDADE", "VL_RENDA_FAMILIAR_BRUTA_MENSAL", "VL_RENDA_PESS
OA_BRUTA_MENSAL",
                                 "VL_RENDA_PERCAPITA", "NU_SEMESTRE_FINANCIADO", "VL_FINANCIA
MENTO")]

cormat <- signif(cor(dffinanum), 2)

plotcorr <- ggcorrplot(cormat, hc.order = TRUE, type = "lower",
           lab =TRUE)
print(plotcorr)
```

Explorando Variáveis Categóricas

Tabelas de Contingências e Proporções de cada categoria

Sexo

```
##str(dffinanciamento)
table(dffinanciamento$SG_SEXO)
round(prop.table(table(dffinanciamento$SG_SEXO)),3)
```

Raça Cor

```
table(dffinanciamento$DS_RACA_COR)
round(prop.table(table(dffinanciamento$DS_RACA_COR)),3)
```

Estado (UF)

```
table(dffinanciamento$SG_UF)
round(prop.table(table(dffinanciamento$SG_UF)),3)
```

Estado Civil

```
table(dffinanciamento$DS_ESTADO_CIVIL)
round(prop.table(table(dffinanciamento$DS_ESTADO_CIVIL)),3)
```

Ensino Médio em Escola Pública

```
table(dffinanciamento$ST_ENSINO_MEDIO_ESCOLA_PUBLICA)
round(prop.table(table(dffinanciamento$ST_ENSINO_MEDIO_ESCOLA_PUBLICA)),3)
```

Bolsista ProUni

```
table(dffinanciamento$ST_BOLSISTA_PROUNI)
round(prop.table(table(dffinanciamento$ST_BOLSISTA_PROUNI)),3)
```

Deficiencia

```
table(dffinanciamento$ST_DEFICIENCIA)
round(prop.table(table(dffinanciamento$ST_DEFICIENCIA)),3)
```

Estado de Inadimplência (variável Target)

```
table(dffinanciamento$ST_INADIMPLENCIA)
round(prop.table(table(dffinanciamento$ST_INADIMPLENCIA)),3)
```

Data Munging

Definindo os fatores para a variável Target e descrevendo o sexo para fácil leitura

```
dffinanciamento$SG_SEXO <- sapply(dffinanciamento$SG_SEXO, function(x){ifelse(x=="F", "Femini
no", "Masculino")})
dffinanciamento$ST_INADIMPLENCIA <- sapply(dffinanciamento$ST_INADIMPLENCIA, function(x){ifel
se(x=="S", 1, 0)})
```

Aplicando Padronização

Criando as funções para transformar variáveis em Fatores e Padronização dos dados

```
convfatores <- function(dataset, variaveis){
  for (variaveis in variaveis){
    dataset[[variaveis]] <- as.factor(dataset[[variaveis]])
  }
  return(dataset)
}

convnormalizacao <- function(dataset, variaveis){
  for (variaveis in variaveis){
    dataset[[variaveis]] <- scale(dataset[[variaveis]], center = T, scale = T)
  }
  return(dataset)
}
```

Visualização dos Datatypes após tratamento

```
##colnames(dffinanciamento)
variaveisnumericas <- c("NU_IDADE", "VL_RENDA_FAMILIAR_BRUTA_MENSAL", "VL_RENDA_PESSOA_BRUTA_
MENSAL",
                        "VL_RENDA_PERCAPITA", "NU_SEMESTRE_FINANCIADO", "VL_FINANCIAMENTO")

variaveiscategoricas <- c('SG_SEXO', 'DS_RACA_COR', 'SG_UF', 'DS_ESTADO_CIVIL', 'ST_DEFICIENC
IA', 'ST_ENSINO_MEDIO_ESCOLA_PUBLICA',
                          'ST_BOLSISTA_PROUNI', 'ST_INADIMPLENCIA')

## Normalização das Variáveis Númericas
dffinanc_normalizado <- convnormalizacao(dffinanciamento, variaveisnumericas)


## Conversão das Variáveis Categóricas em Fatores
dffinanciamentoconv <- convfatores(dffinanc_normalizado, variaveiscategoricas)

## Confirmando as transformações
str(dffinanciamentoconv)
```

Preparando os dados para Treino e Teste

```
set.seed(2021)
indicetreino <- sample(1:nrow(dffinanciamentoconv), 0.7 * nrow(dffinanciamentoconv))
dftreino <- dffinanciamentoconv[indicetreino, ]

indiceteste <- c(1:nrow(dffinanciamentoconv))[-indicetreino]
dfteste <- dffinanciamentoconv[indiceteste, ]

#3class(dftreino)
##class(dfteste)
```

Sumário da Base de Treino

```
summary(dftreino)
```

– As distribuições das variáveis categórica com a Variável Target estão OK.

Construindo o Modelo de Regressão Logística com Padronização

```
## Do lado esquerdo do '~' a variável a ser prevista, do lado direito, as variáveis preditoras.
## O '.' representa todas as variáveis do Dataset
## binomial por ser variável binária
modeloreglog <- glm(as.formula("ST_INADIMPLENCIA ~ ."), data = dftreino, family = "binomial"(link="logit"))
summary(modeloreglog)
```

###Glossário sobre o sumário acima:

####Resíduos: Diferença entre os valores observados e os valores previstos. ##### Devem se parecer com uma distribuição normal, o que indica que a média entre #### os valores previstos e os valores observados é próximo de 0.

####Erro Padrão: Medida de variabilidade na estimativa do coeficiente. O ideal é que este valor seja menor que o valor do coeficiente.

Construindo o Modelo de Previsões

```
previsoes <- predict(modeloreglog, dfteste, type = "response")
previsoes <- round(previsoes)
previsoes <- as.factor(previsoes)
```

Visualização do Resultado do modelo : Matriz de Confusão e Resumo Estatístico

```
variaveisteste <- dfteste[, -14]
targetteste <- dfteste[, c("ST_INADIMPLENCIA")]

##class(variaveisteste)
##class(targetteste)

confusionMatrix(table(data = previsoes, reference = targetteste), positive = '1')
```

Visualização da Curva do Modelo de Regressão com Normalização

```
fit_glm <- glm("ST_INADIMPLENCIA ~ .", dftreino, family = binomial(link="logit"))
glm_link_score <- predict(fit_glm, dfteste, type="link")
glm_response_score <- predict(fit_glm, dfteste, type = "response")
score_data <- data.frame(link = glm_link_score,
                         response=glm_response_score,
                         ST_INADIMPLENCIA=dfteste$ST_INADIMPLENCIA,
                         stringsAsFactors = FALSE)


score_data %>%
  ggplot(aes(x=link, y=response, col=ST_INADIMPLENCIA)) +
  scale_color_manual(values=c("blue", "red")) +
  geom_point() +
  geom_rug() +
  ggtitle("Plot do Modelo Preditivo")

plot(roc(dfteste$ST_INADIMPLENCIA, glm_response_score, direction="<"),
     col="blue", lwd=3, main="Plot do Resultado do Modelo de Previsão com Normalização")
```

Visualização dos desvios de cada variável no modelo com ANOVA

```
anova(modeloreglog, test="Chisq")
```

####Ao analisar os desvios das variáveis, verificamos que para esse modelo as variáveis “DS_ESTADO_CIVIL”, “ST_DEFICIENCIA”, “VALOR_RENDA_PESSOAL_BRUTA” e “NU_SEMESTRE FINANCIADO” aparentam melhorar menos o modelo, embora “nu_semestre_financiado” esteja com valor p inferior a 0.05.

Aplicando Regularização LASSO

```
##Recriando as bases de Treino e Teste sem Normalização
set.seed(2021)
idx_dftreinolasso <- sample(1:nrow(dffinanciamento), 0.7 * nrow(dffinanciamento))
dftreinolasso <- dffinanciamento[idx_dftreinolasso,]

idx_dftestelasso <- c(1:nrow(dffinanciamento))[-idx_dftreinolasso]
dftestelasso <- dffinanciamento[idx_dftestelasso,]

## Variáveis Target
targettreinolasso <- dftreinolasso[, c("ST_INADIMPLENCIA")]
targettestelasso <- dftestelasso[, c("ST_INADIMPLENCIA")]


## Criando a Matrix de variáveis
matrixbasetreino <- data.matrix(dftreinolasso[, -ncol(dftreinolasso)])
matrixbaseteste <- data.matrix(dftestelasso[, -ncol(dftestelasso)])


## Descobrindo o melhor Lambda
set.seed(2021)
modelocvlasso <- cv.glmnet(matrixbasetreino, targettreinolasso, alpha = 1)
minlambda <- modelocvlasso$lambda.min
minlambda


## Melhor Coeficiente
melhormodelocvlasso <- glmnet(matrixbasetreino, targettreinolasso, alpha = 1, lambda = minlam
bda)
coef(melhormodelocvlasso)
```

Criando o modelo de Previsão com Lasso

```
previsaolasso1 <- predict(melhormodelocvlasso, newx = matrixbaseteste, s = minlambda)

## Desempenho do Modelo usando MSE
MSElasso1 <- mean((previsaolasso1 - targettestelasso)^2)
MSElasso1
```

Comparando os MSE dos modelos LASSO x Regressão Logística

```
modeloregloglasso <- glm(as.formula("ST_INADIMPLENCIA ~ ."), data = dftreinolasso, family = b
inomial(link="logit"))
previsoeslasso <- predict(modeloregloglasso, dftestelasso, type = "response")
previsoeslasso <- round(previsoeslasso)

MSEprevisao1 <- mean((previsoeslasso - targettestelasso)^2)
c(MSElasso1, MSEprevisao1)
```

Visualização do Resultado do modelo : Matriz de Confusão e Resumo Estatístico

```
confusionMatrix(table(data = as.factor(previsoeslasso), reference = as.factor(dftestelasso$ST
_INADIMPLENCIA)), positive = '1')
```

###O modelo LASSO demonstrou melhor performance em relação ao modelo de regressão logística. ###Com o cálculo de coeficiente de cada variável, podemos selecionar as variáveis com valores mais próxios de 0 para testar um novo modelo (feature selection por LASSO).

Feature Selection

Utilizando K-Fold Cross Validation

```
formula <- as.formula("ST_INADIMPLENCIA ~ .")
controle <- trainControl(method = "repeatedcv", number = 6, repeats = 3)
modelocontrole <- train(formula, data = dftreino, method="glm", trControl = controle)
importance <- varImp(modelocontrole, scale = FALSE)
plot(importance)
```

Testando o modelo LASSO com seleção de variáveis

```
## Removação de outliers pela formula Q3 + 1.5 * IQR
##Recriando as bases de Treino e Teste sem Normalização
set.seed(2021)
dflasso <- dffinanciamento

## Removendo as colunas multicolinearidade (renda bruta) e sem efeito para modelo (st_deficie
ncia)
dflasso$VL_RENDA_PESSOA_BRUTA_MENSAL <- NULL
dflasso$ST_DEFICIENCIA <- NULL

## Removendo os Outliers
Q3capitalasso <- quantile(dflasso$VL_RENDA_PERCAPITA, probs = 0.75)
IQRcapitalasso <- IQR(dflasso$VL_RENDA_PERCAPITA)
formulaoutlierscapitalasso <- Q3capitalasso + 1.5 * IQRcapitalasso
dflassofilter <- filter(dflasso, VL_RENDA_PERCAPITA <= formulaoutlierscapitalasso)


idx_dftreinolassofilter <- sample(1:nrow(dflassofilter), 0.7 * nrow(dflassofilter))
dftreinolassofilter <- dflassofilter[idx_dftreinolassofilter,]

idx_dftestelassofilter <- c(1:nrow(dflassofilter))[-idx_dftreinolassofilter]
dftestelassofilter <- dflassofilter[idx_dftestelassofilter,]

## Variáveis Target
targettreinolassofilter <- dftreinolassofilter[, c("ST_INADIMPLENCIA")]
targettestelassofilter <- dftestelassofilter[, c("ST_INADIMPLENCIA")]


## Criando a Matrix de variáveis
matrixbasetreinofilter <- data.matrix(dftreinolassofilter[, c("VL_RENDA_FAMILIAR_BRUTA_MENSA
L",
                              "ST_ENSINO_MEDIO_ESCOLA_PUBLICA",
                              "DS_ESTADO_CIVIL",
                              "VL_FINANCIAMENTO",
                              "SG_SEXO",
                              "ST_BOLSISTA_PROUNI")])

matrixbasetestefilter <- data.matrix(dftestelassofilter[, c("VL_RENDA_FAMILIAR_BRUTA_MENSAL",
                              "ST_ENSINO_MEDIO_ESCOLA_PUBLICA",
                              "DS_ESTADO_CIVIL",
                              "VL_FINANCIAMENTO",
                              "SG_SEXO",
                              "ST_BOLSISTA_PROUNI")])

## Descobrindo o melhor Lambda
set.seed(2021)
modelocvlassofilter <- cv.glmnet(matrixbasetreinofilter, targettreinolassofilter, alpha = 1)
minlambdafilter <- modelocvlassofilter$lambda.min
minlambdafilter


## Melhor Coeficiente
melhormodelocvlassofilter <- glmnet(matrixbasetreinofilter, targettreinolassofilter, alpha =
1, lambda = minlambdafilter)
coef(melhormodelocvlassofilter)
```

Criando o modelo de Previsão com Lasso - com seleção de variáveis

```
previsaolassofilter<-  predict(melhormodelocvlassofilter,
                               newx = matrixbasetestefilter,
                               s = minlambdafilter)

## Desempenho do Modelo usando MSE
MSElassofilter <- mean((previsaolassofilter - targettestelassofilter)^2)
MSElassofilter
```

A seleção de variáveis para o modelo LASSO não foi efetiva, tendo aumentado o MSE após a seleção.

Removendo outliers para Otimização do Modelo após Padronização

```
## Removação de outliers pela formula Q3 + 1.5 * IQR
Q3capita <- quantile(dffinanciamentoconv$VL_RENDA_PERCAPITA, probs = 0.75)
IQRcapita <- IQR(dffinanciamentoconv$VL_RENDA_PERCAPITA)
formulaoutlierscapita <- Q3capita + 1.5 * IQRcapita
dffinanciamentoconvcapita <- filter(dffinanciamentoconv, VL_RENDA_PERCAPITA <= formulaoutlier
scapita)

## Removendo as colunas multicolinearidade (renda bruta) e sem efeito para modelo (st_deficie
ncia)
dffinanciamentoconvcapita$VL_RENDA_PESSOA_BRUTA_MENSAL <- NULL
dffinanciamentoconvcapita$ST_DEFICIENCIA <- NULL

## Demonstração quantitativa e estatísitca das remoções
table(dffinanciamento$ST_INADIMPLENCIA)
table(dffinanciamentoconvcapita$ST_INADIMPLENCIA)
summary(dffinanciamentoconvcapita)
```

Feature Selection após remoção dos Outliers

```
## Feature Selection após remoção das variáveis
formula <- as.formula("ST_INADIMPLENCIA ~ .")
controle <- trainControl(method = "repeatedcv", number = 6, repeats = 3)
modelocontrole <- train(formula, data = dffinanciamentoconvcapita, method="glm", trControl =
 controle)
importance <- varImp(modelocontrole, scale = FALSE)
plot(importance)
```

Treinando o Modelo de Regressão com Seleção de Variáveis

```
## Separando em dados de Treino e Teste
set.seed(2021)
indicetreinocapita <- sample(1:nrow(dffinanciamentoconvcapita), 0.7 * nrow(dffinanciamentocon
vcapita))
dftreinocapita <- dffinanciamentoconvcapita[indicetreinocapita, ]

indicetestecapita <- c(1:nrow(dffinanciamentoconvcapita))[-indicetreinocapita]
dftestecapita <- dffinanciamentoconvcapita[indicetestecapita, ]


## Treinando o modelo
modeloreglog4 <- glm(ST_INADIMPLENCIA ~ VL_RENDA_FAMILIAR_BRUTA_MENSAL
                     + ST_ENSINO_MEDIO_ESCOLA_PUBLICA
                     + DS_ESTADO_CIVIL
                     + VL_FINANCIAMENTO
                     + SG_SEXO
                     + ST_BOLSISTA_PROUNI,
                     data = dftreinocapita,
                     family = "binomial")
summary(modeloreglog4)
```

Visualização do Resultado do modelo : Matriz de Confusão e Resumo Estatístico

```
## Fazendo as previsões do modelo
previsoes4 <- predict(modeloreglog4, dftestecapita, type = "response")
previsoes4 <- round(previsoes4)
previsoes4 <- as.factor(previsoes4)

targettestecapita <- dftestecapita[, c("ST_INADIMPLENCIA")]

confusionMatrix(table(data = previsoes4, reference = targettestecapita), positive = '1')
```

Visualização do Resultado do modelo : Matriz de Confusão e Resumo Estatístico

Testando com a base de Outliers

```
dfoutliers <- filter(dffinanciamentoconv, VL_RENDA_PERCAPITA > formulaoutlierscapita)
previsaooutliers4 <- predict(modeloreglog4, dfoutliers, type = "response")
previsaooutliers4 <- round(previsaooutliers4)
previsaooutliers4 <- as.factor(previsaooutliers4)

confusionMatrix(previsaooutliers4, dfoutliers$ST_INADIMPLENCIA, positive = "1")
```

Análise Preditiva com Árvore de Decisão

Testando um modelo de árvore de Decisão - com seleção de variáveis (sem excluir outliers)**

```
library(rpart)

## Removendo as colunas
dftreino$VL_RENDA_PESSOA_BRUTA_MENSAL <- NULL
dftreino$ST_DEFICIENCIA <- NULL
dfteste$VL_RENDA_PESSOA_BRUTA_MENSAL <- NULL
dfteste$ST_DEFICIENCIA <- NULL

## Treinando o modelo com a seleção de variável testada no modelo de Regressão Logística
modelorf <- rpart(ST_INADIMPLENCIA ~ VL_RENDA_FAMILIAR_BRUTA_MENSAL
                     + ST_ENSINO_MEDIO_ESCOLA_PUBLICA
                     + DS_ESTADO_CIVIL
                     + VL_FINANCIAMENTO
                     + SG_SEXO
                     + ST_BOLSISTA_PROUNI,
                  data = dftreino,
                  control = rpart.control(cp = 0.05))
```

Acurácia do Modelo de Árvore de Decisão com Rpart

```
## Fazendo as previsões do modelo
treepred <- predict(modelorf, dfteste, type = "class")
mean(treepred==dfteste$ST_INADIMPLENCIA)
```

Matrix de confusão

```
table(treepred, dfteste$ST_INADIMPLENCIA)
```

Sensibilidade do Modelo de Árvore de Decisão com Rpart

```
sensitivity(treepred, reference = dfteste$ST_INADIMPLENCIA, positive = '1')
```

Códigos que não rodaram

Aplicando Cost Function - penalizar o modelo para erro de Falso Negativo (direcionado pelo trabalho)

```
library(C50)
##w21 : o peso está de 2 para previsões de Falso Negativo
costfunc <- matrix(c(0, 2 , 1, 0), nrow = 2, dimnames = list(c("1", "2"), c("1", "2")))

modelocostfunc <-C5.0(ST_INADIMPLENCIA ~ ., data = dftreino, rules = TRUE, control = C5.0Cont
rol(), cost = costfunc)
print(modelocostfunc)

previsoes2 <- predict(object = modelocostfunc, newdata = dfteste, type = "class", na.action=
 na.pass)

confusionMatrix(table(data = previsoes2, reference = targetteste), positive = "1")
```

RandomForest por função

```
rffunc <- randomForest(ST_INADIMPLENCIA ~ .,
                       data = dftreino,
                       importance = TRUE,
                       proximity = TRUE)

predmodelrf <- predict(rffunc, dfteste)
mean(predmodelrf==dfteste$ST_INADIMPLENCIA)

table(predmodelrf, dfteste$ST_INADIMPLENCIA)
summary(predmodelrf)
```

```
library(mlbench)
## Separando em dados de Treino e Teste
indicetreinocost <- sample(1:nrow(dffinanciamentoconv), 0.7 * nrow(dffinanciamentoconv))
dftreinocost <- dffinanciamentoconv[indicetreinocost, ]

indicetestecost <- c(1:nrow(dffinanciamentoconv))[-indicetreinocost]
dftestecost <- dffinanciamentoconv[indicetestecost, ]


fitcontrol <- trainControl(method="repeatedcv", number=10, repeats=5)

##w21 : o peso está de 2 para previsões de Falso Negativo
costfunc <- matrix(c(0, 2 , 1, 0), nrow = 2, dimnames = list(c("1", "2"), c("1", "2")))

statGrid <- expand.grid(trials = 3,
                        model = "tree",
                        winnow = FALSE,
                        cost = costfunc)

statFit <- train(ST_INADIMPLENCIA ~ VL_RENDA_FAMILIAR_BRUTA_MENSAL
                 + VL_RENDA_PERCAPITA
                 + ST_ENSINO_MEDIO_ESCOLA_PUBLICA
                 + VL_FINANCIAMENTO
                 + ST_BOLSISTA_PROUNI
                 + DS_ESTADO_CIVIL
                 + NU_IDADE
                 + DS_REGIAO,
                 data = dftreinocost,
                 method = "C5.0Cost",
                 trControl = fitcontrol,
                 tuneGrid = statGrid,
                 metric = "Accuracy")
```

Construindo o modelo SVM

```
## Criando o índice para divisao de dados de treino e de teste
dadosdf <- dffinanciamento
dadosdf[, 'indices'] <- ifelse(runif(nrow(dadosdf)) < 0.7, 1, 0)

## Saparando os dados
dadostreino <- filter(dadosdf, indices == 1)
dadosteste <- filter(dadosdf, indices != 1)

## Gera o índice das linhas
linhaindice <- grep('indices', names(dadostreino))

## Removendo o Índice dos dados de treino e de teste
dadostreino <- dadostreino[, -linhaindice]
dadosteste <- dadosteste[, -linhaindice]

## Gera o índice das colunas
colunaindice <- grep('ST_INADIMPLENCIA', names(dadosdf))

## Construindo o Modelo de Predição
library(e1071)
modelosvm <- svm(ST_INADIMPLENCIA ~ . ,
                 data = dadostreino ,
                 type = 'C-classification',
                 kernel = 'radial')
```

```
previsaosvm <- predict(modelosvm, dadosteste)

mean(previsaosm == dadosteste$ST_INADIMPLENCIA)

## Confusion Matrix
table(previsaosvm, dadosteste$ST-INADIMPLENCIA)
```