Atividade 1 - Regressão Logística Binária

Grupo 404 (Guilherme Orlandi, Luis Felipe Degaspari, Luiz Felipe Pimenta e Luiz Fernando Pazdziora)

2025-09-21

Regressão Logística Binária para Banco de Varejo

Um grande banco de varejo deseja melhorar sua política de concessão de crédito. Para isso, reuniu um conjunto de dados com informações históricas sobre seus clientes e os empréstimos concedidos.

Cada registro representa um empréstimo individual, associado a dados demográficos e financeiros do cliente, tais como renda, histórico de crédito, situação de moradia, tempo de emprego e pontuação de crédito.

O objetivo do banco é construir um modelo preditivo capaz de estimar a probabilidade de inadimplência em novos pedidos de crédito.

Variáveis da base

- Loan Status → Situação do empréstimo (adimplente ou inadimplente, que estão na base como 0 e 1, respectivamente).
- Credit Score → Pontuação de crédito do cliente (quanto maior a pontuação, menor o risco de inadimplência).
- Annual Income → Renda anual (em dólares).
- Home Ownership → Situação da moradia (própria, alugada ou hipotecada, que estão na base como Own Home, Rent e Home Mortgage, respectivamente).
- Purpose → Finalidade do empréstimo (dívida, empréstimo para negócios, reformas na casa, pequeno negócio, compra de casa, outro, casamento, despesas médicas, compra de carro, mudança, viagem, grande compra, férias ou despesas educacionais, que estão na base como Debt Consolidation, Business Loan, Home Improvements, small_business, Buy House, Other, wedding, Medical Bills, Buy a Car, moving, Take a Trip, major_purchase, vacation e Educational Expenses, respectivamente).
- Tax Liens → Número de gravames fiscais (bens que estão como garantia de uma dívida).

Agora, temos que:

- Variável Dependente: Loan Status
- Variáveis Independentes: Credit Score, Annual Income, Home Ownership, Purpose, Tax Liens

Criação das dummies

Agora, vamos criar dummies para as variáveis categóricas `Home Ownership` e `Purpose`, dividindo cada um dos seus possíveis valores em diferentes colunas com valores binários. Para `Home Ownership`, a referência é "Own Home", e para `Purpose`, a referência é "Medical Bills".

As linhas de código abaixo criam as dummies:

```r
dados$`Home Ownership`=factor(dados$`Home Ownership`, levels=c("Own Home", "Home Mortgage", "Rent"))

dados$Purpose = factor(dados$Purpose, levels = c("Medical Bills", "Debt Consolidation", "Business Loan"
```

Separação da base de dados em conjunto de treino e teste

O próximo passo é dividir os dados em dois conjuntos distintos: o **conjunto de treino** e o **conjunto de teste**. O conjunto de treino será utilizado para treinar o modelo de machine learning. O conjunto de teste será utilizado para testar a precisão do modelo.

A divisão será de 70% dos dados originais para treino e os 30% restantes para teste.

```r
set.seed(123)
flag = createDataPartition(dados$`Loan Status`, p= 0.7, list = F)
# p definido como 0.7 (70% da base será utilizada como treino)

# Criando a base de dados de treino
train = dados[flag,]

# Criando a base de dados de teste
test = dados[-flag,]

kable(round(table(train$`Loan Status`), 2), caption = "Quantidade de registros na amostra de treino")
```

Table 1: Quantidade de registros na amostra de treino

Var1	Freq
0	20274
1	5220

```r
kable(round(table(test$`Loan Status`), 2), caption = "Quantidade de registros na amostra de teste")
```

Table 2: Quantidade de registros na amostra de teste

Var1	Freq
0	8694
1	2231

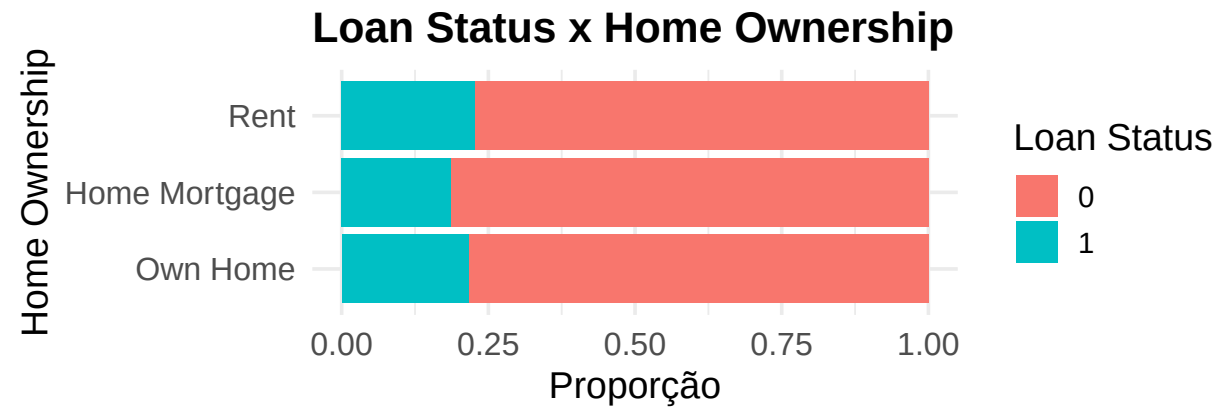
Análise descritiva do conjunto de treino

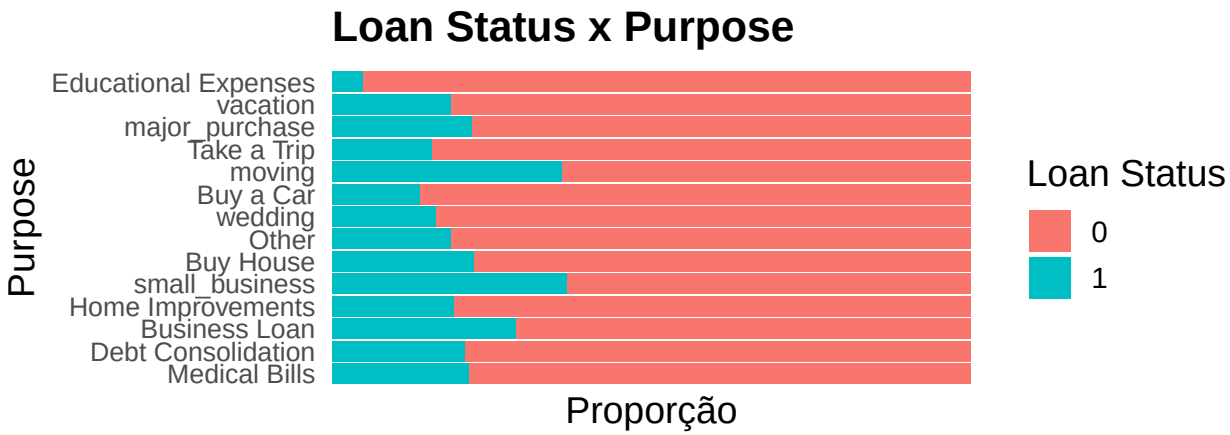
Primeiramente, vamos analisar as variáveis qualitativas que foram transformadas em dummies. Para entender se elas podem impactar na proporção de `Status Loan`, ou seja, se ela consegue diferenciar os adimplentes e inadimplentes, vamos realizar gráficos de barras empilhadas.

Variáveis `Home Ownership` e `Purpose`

```
## Warning in element_text(size = 16, face = "bold", axis.text.y = element_text(size = 12, : `...` must
## x Problematic arguments:
## * axis.text.y = element_text(size = 12, margin = margin(r = 10))
## * plot.title = element_text(size = 16, face = "bold")
## * plot.margin = margin(30, 30, 30, 60)
```

```
## Warning in element_text(size = 16, face = "bold", axis.text.y = element_text(size = 12, : `...` must
## x Problematic arguments:
## * axis.text.y = element_text(size = 12, margin = margin(r = 10))
## * plot.title = element_text(size = 16, face = "bold")
## * plot.margin = margin(0, 0, 0, 0)
```



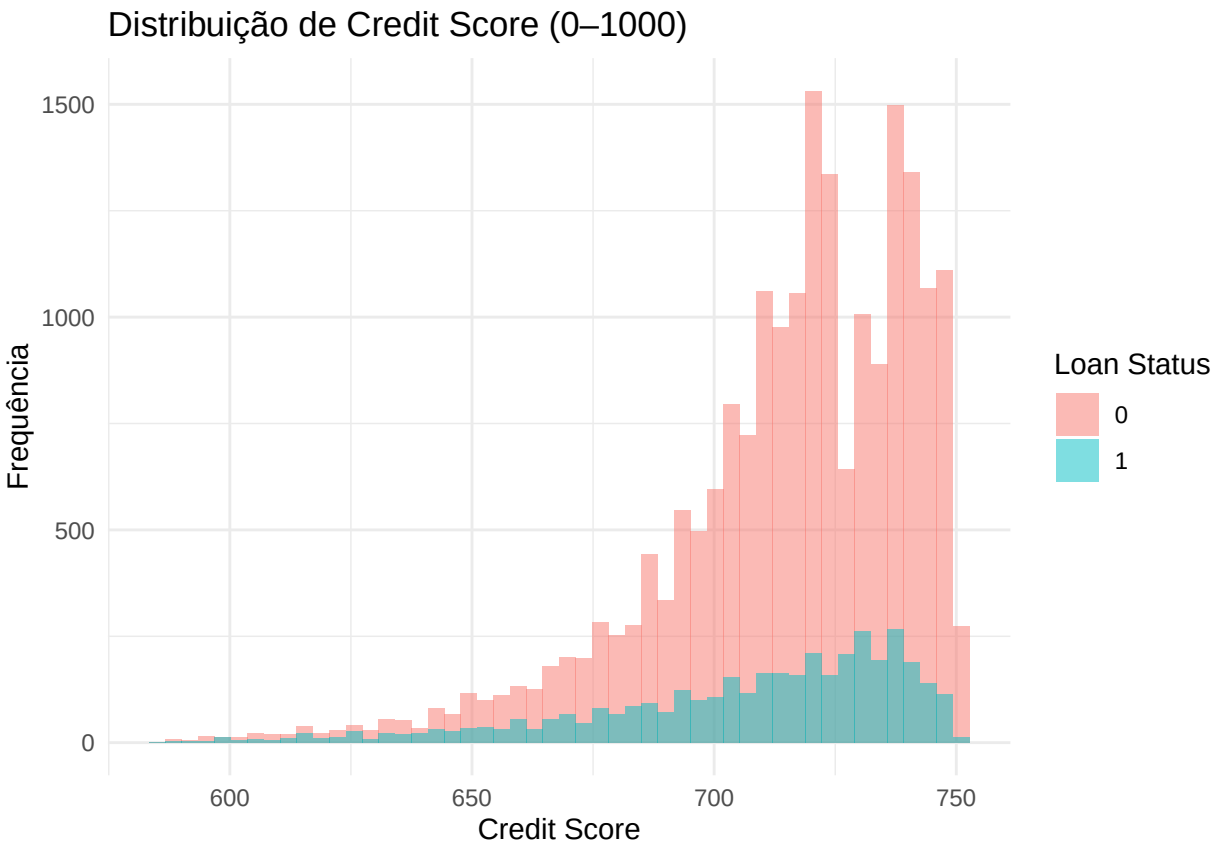


A variável `Home Ownership` parece não explicar o `Loan Status`, já que não há grandes variações na proporção de adimplentes e inadimplentes de acordo com a mudança de categorias.

Já a variável `Purpose` possui uma variação da proporção entre adimplentes e inadimplentes significante de acordo com cada categoria, indicando que ela possui correlação com a variável dependente.

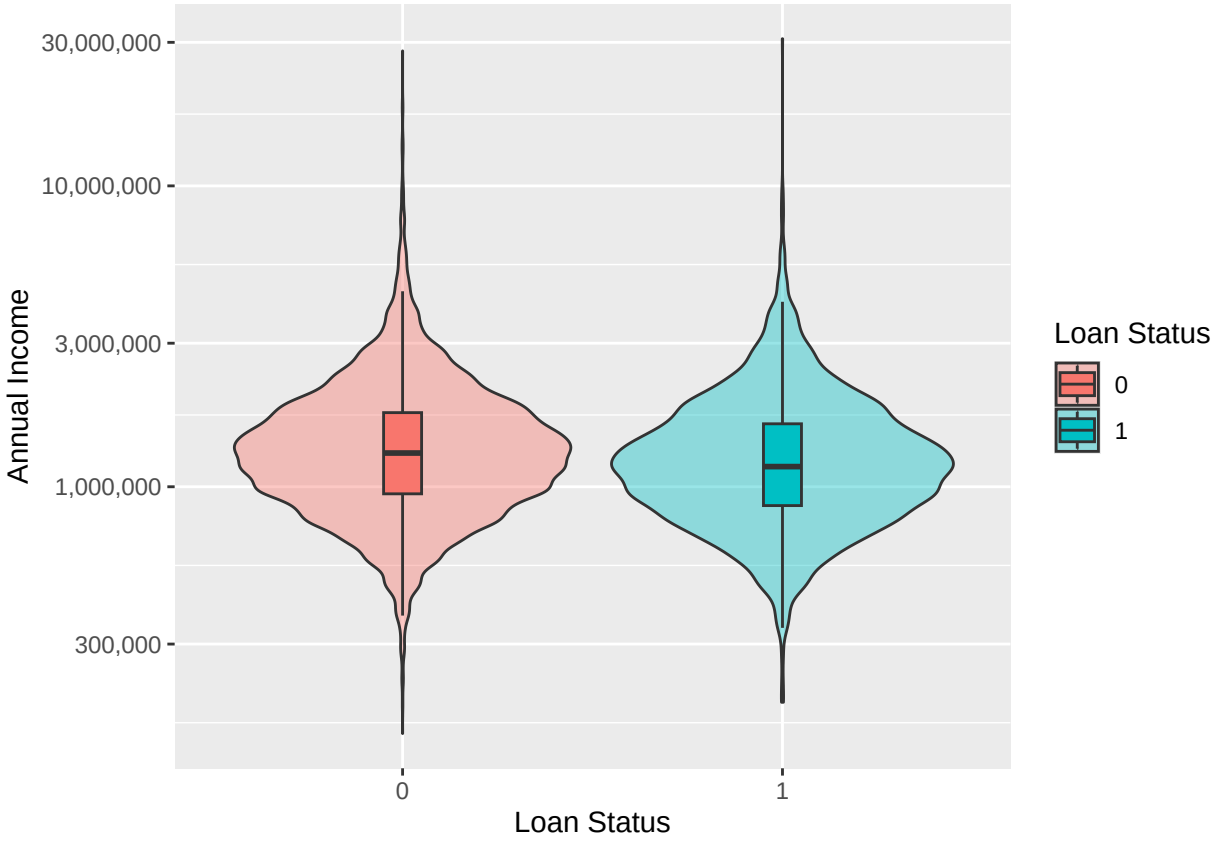
Em seguida, iremos analisar as variáveis explicativas quantitativas. Para esse tipo de variável, iremos realizar dois gráficos: Um histograma e um violino + boxplot. Ambos fazem a comparação entre os valores das variáveis independentes quantitativas quando o registro é adimplente ou inadimplente. A comparação da proporção entre os valores no caso de `Loan Status` ser 0 ou 1 que vai nos auxiliar a entender se a variável pode explicar, ou não, a variável-alvo.

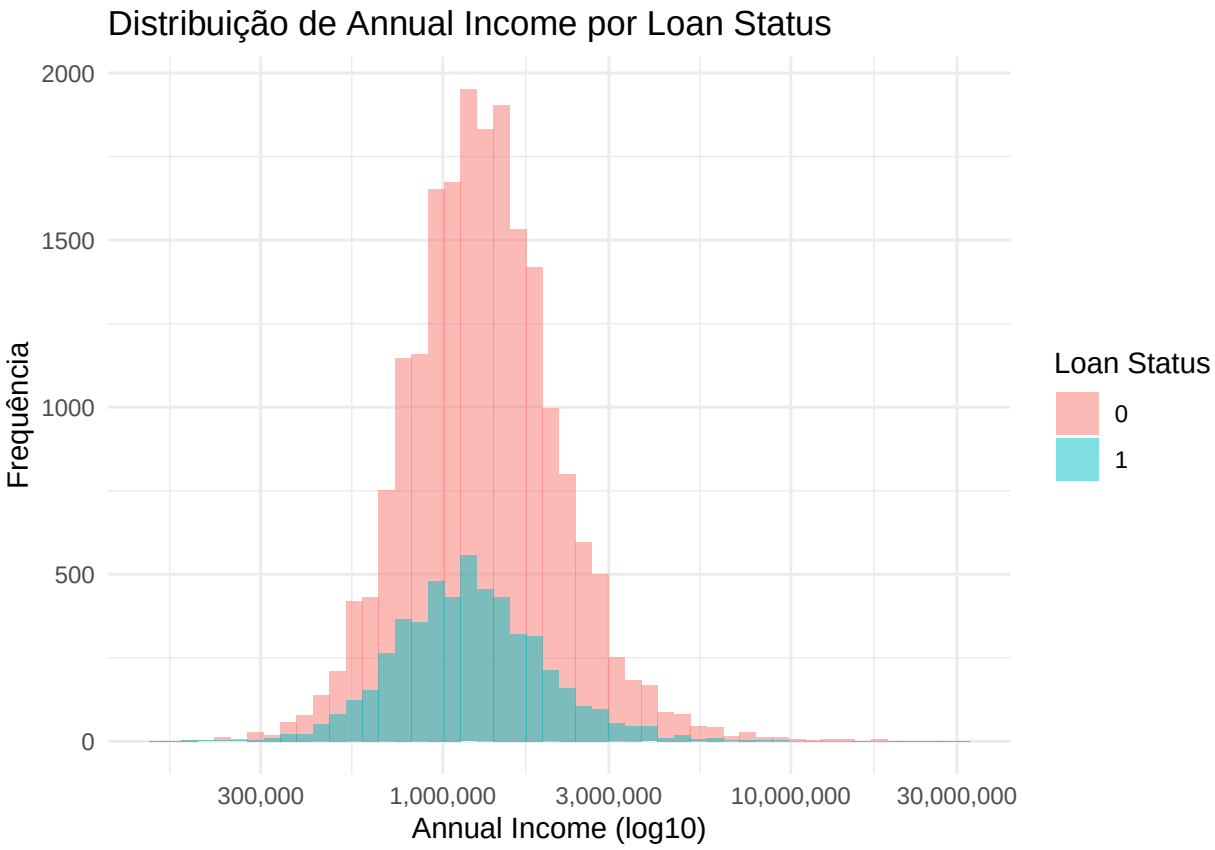
Variável `Credit Score`



A variável `Credit Score` possui um grande número de *outliers*. Portanto, foi feita uma redução da faixa de observação para valores da variável independente entre 0-1000, apenas para possibilitar a observação das curvas dos histogramas e compará-las. Aqui, é possível notar que a frequência é proporcional quando `Loan Status` é 0 e 1, indicando que a variável `Credit Score` não possui capacidade de explicação da variável dependente.

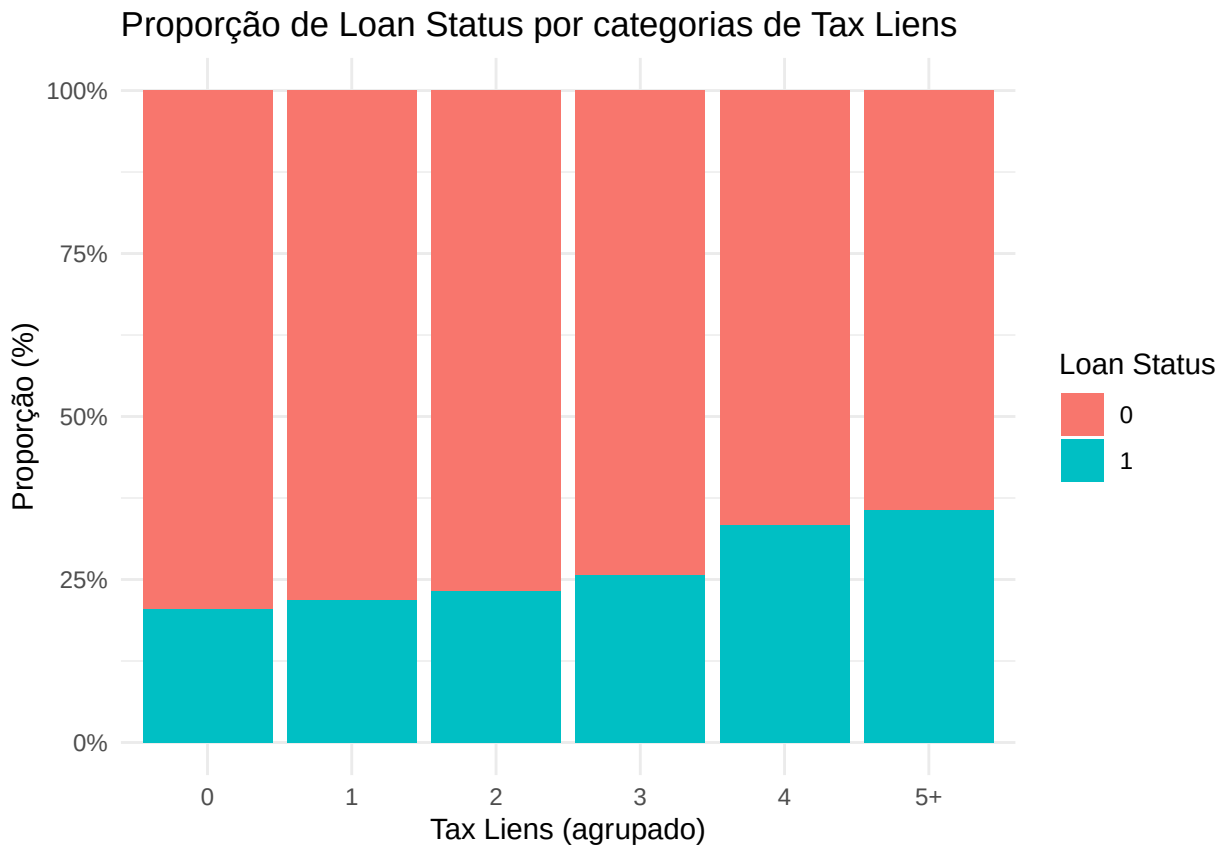
Variável `Annual Income`





As visualizações esclarecem que, independendo de `Loan Status` ser 0 ou 1, ou seja, o cliente ser adimplente ou inadimplente, os valores dessas variáveis se mantém semelhantes. Esse fator indica que, provavelmente, a variável `Annual Income` não possui poder explicativo sobre a variável-alvo.

Variável `Tax Liens`



Para essa variável, utilizamos o gráfico de barras empilhadas devido aos *outliers* presentes, que danificavam a visualização no boxplot e histograma. Além disso, agrupamos todos os valores de `Tax Liens` que fossem maiores ou iguais a 5, pois os registros ficam escassos nessa faixa.

É possível observar que, devido à pequena variância no gráfico de barras empilhadas, a variável `Tax Liens` possui pouca capacidade de explicação da variável dependente `Loan Status`. A relação é diretamente proporcional, indicando que quanto maior o valor de `Tax Liens`, maior a chance do cliente ser inadimplente.

Verificação de multicolinearidade

O próximo passo é checar se as variáveis do modelo possuem multicolinearidade, ou seja, se as variáveis independentes do modelo estão correlacionadas entre si. Caso estejam correlacionadas, devemos removê-las.

Em seguida, vamos calcular o VIF (Variance Influence Factor) de cada uma das variáveis independetes do modelo. Caso algum valor seja maior do 5, removeremos as variáveis uma a uma.

Table 3: Vifs do modelo1

	GVIF	Df	GVIF^(1/(2*Df))
`Credit Score`	1.01	1	1.00
`Annual Income`	1.05	1	1.02

	GVIF	Df	GVIF^(1/(2*Df))
Tax Liens	1.00	1	1.00
Home Ownership	1.07	2	1.02
Purpose	1.05	13	1.00

Como é possível observar pela tabela, os VIFs estão regulares, com nenuhum valor acima de 5. Portanto, não há multicolinearidade no modelo.

Modelo Logístico Preliminar

Em seguida, vamos realizar o modelo logístico preliminar, antes de realizar o stepwise nas variáveis:

Table 4: Modelo logístico preliminar

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.13	0.18	-11.80	0.00
Credit Score	0.00	0.00	15.52	0.00
Annual Income	0.00	0.00	-8.31	0.00
Tax Liens	0.11	0.05	2.20	0.03
Home OwnershipHome Mortgage	-0.17	0.06	-2.60	0.01
Home OwnershipRent	0.05	0.06	0.70	0.48
PurposeDebt Consolidation	-0.04	0.16	-0.22	0.82
PurposeBusiness Loan	0.53	0.20	2.60	0.01
PurposeHome Improvements	-0.04	0.18	-0.22	0.83
Purposesmall_business	0.95	0.30	3.20	0.00
PurposeBuy House	-0.01	0.26	-0.03	0.98
PurposeOther	-0.14	0.17	-0.84	0.40
Purposewedding	-0.08	0.52	-0.15	0.88
PurposeBuy a Car	-0.66	0.26	-2.54	0.01
Purposemoving	0.80	0.40	2.01	0.04
PurposeTake a Trip	-0.41	0.29	-1.41	0.16
Purposemajor_purchase	-0.10	0.37	-0.27	0.79
Purposevacation	-0.22	0.57	-0.38	0.70
PurposeEducational Expenses	-1.49	1.04	-1.43	0.15

O modelo logístico preliminar fornece as estimativas dos coeficientes, o desvio-padrão e valor-p de todas as variáveis do modelo. Grande parte das variáveis no modelo logístico preliminar provém da criação das dummies.

Com esses dados, é possível observar que a variável `Purposesmall_bussiness` é a que possui maior capacidade de explicação da variável dependente, já que possui o maior coeficiente. Além disso, variáveis como `Purposewedding` e `Purposemajor_purchase` possuem valor-p extremamente alto, significando que possuem pouquíssima ou nenhuma correlação com a variável-alvo `Loan Status`.

Com isso, temos:

- Variáveis com valor-p maior do que 5% (pouco ou não correlacionadas com a variável dependente): `Home OwnershipRent`, `PurposeDebt Consolidation`, `PurposeHome Improvements`, `PurposeBuy House`, `PurposeOther`, `Purposewedding`, `PurposeTake a Trip`, `Purposemajor_purchase`, `Purposevacation` e `PurposeEducational Expenses`.
- Variáveis com coeficiente extremamente próximo de 0 (Alteram muito minimamente o resultado da predição e, quando arredondadas para duas casas decimais, ficam completamente zeradas): `Credit`

Score, Annual Income, PurposeDebt Consolidation, PurposeHome Improvements e PurposeBuy House.

Modelo Logístico Ótimo

Table 5: Modelo logístico ótimo

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.13	0.18	-11.80	0.00
Credit Score	0.00	0.00	15.52	0.00
Annual Income	0.00	0.00	-8.31	0.00
Tax Liens	0.11	0.05	2.20	0.03
Home OwnershipHome Mortgage	-0.17	0.06	-2.60	0.01
Home OwnershipRent	0.05	0.06	0.70	0.48
PurposeDebt Consolidation	-0.04	0.16	-0.22	0.82
PurposeBusiness Loan	0.53	0.20	2.60	0.01
PurposeHome Improvements	-0.04	0.18	-0.22	0.83
Purposesmall_business	0.95	0.30	3.20	0.00
PurposeBuy House	-0.01	0.26	-0.03	0.98
PurposeOther	-0.14	0.17	-0.84	0.40
Purposewedding	-0.08	0.52	-0.15	0.88
PurposeBuy a Car	-0.66	0.26	-2.54	0.01
Purposemoving	0.80	0.40	2.01	0.04
PurposeTake a Trip	-0.41	0.29	-1.41	0.16
Purposemajor_purchase	-0.10	0.37	-0.27	0.79
Purposevacation	-0.22	0.57	-0.38	0.70
PurposeEducational Expenses	-1.49	1.04	-1.43	0.15

O step realizado no modelo não retirou nenhuma variável, mesmo que hajam diversas variáveis com coeficientes quase nulos e valores-p altos. Isso acontece pois nenhuma variável proporcionou redução do AIC ao ser retirada. O step() tem um critério global, não local, e por isso as variáveis com valor-p alto também não foram retiradas. Isso significa que, mesmo não sendo significativas individualmente, a presença dessas variáveis contribui ligeiramente para o ajuste global medido pelo AIC.

Portanto, a interpretação dos coeficientes permanece igual em relação ao modelo preliminar. Há variáveis que possuem coeficiente muito próximo de zero, o suficiente para serem completamente zeradas no arredondamento para duas casas decimais, como Credit Score e Annual Income, e a variável com maior poder de explicação é Purposesmall_business, significando que a proposta do cliente de ter um pequeno comércio é o fator que mais afeta ele ser adimplente ou inadimplente.

Com isso, temos:

- Variáveis com maior capacidade de explicação: Purposesmall_bussiness, Purposemoving e PurposeEducational Expenses.
- Variáveis com coeficiente muito próximo de 0: Credit Score, Annual Income, PurposeDebt Consolidation, PurposeHome Improvements e PurposeBuy House.
- Variáveis que possuem relação diretamente proporcional com Loan Status: Tax Liens, Home OwnershipRent, PurposeBusiness Loan, Purposesmall_business e Purposemoving.
- Variáveis que possuem relação inversamente proporcional com Loan Status: Home OwnershipHome Mertgage, PurposeOther, Purposewedding, PurposeBuy a Car, PurposeTake a Trip, Purposemajor_purchase, Purposevacation e PurposeEducational Expenses.

Teste do modelo

Em seguida, vamos fazer a previsão da probabilidade de um cliente ser inadimplente e realizar a classificação utilizando o ponto de corte de 50%. O código abaixo realiza essas etapas:

```
test$pstatus = predict(modelo_otimo, newdata = test,
type='response')

pcorte =0.5
test$classificacao = ifelse(test$pstatus > pcorte, 1 , 0)
```

Matriz de Confusão

Agora, vamos analisar o resultado da predição feita. A matriz de confusão vai demonstrar valores previstos vs valores reais e será possível analisar o modelo.

Table 6: Matriz de Confusão - Previsto vs Real

	Real: 0	Real: 1
Previsto: 0	8694	1642
Previsto: 1	0	589

A matriz de confusão evidencia alguns problema no modelo. Vamos analisá-los um a um:

- Alto número de falsos negativos: A alta quantidade de valores previstos como negativos que são, na verdade, positivos, indica que muitos inadimplentes não estão sendo detectados pelo modelo com probabilidade de corte de 50%. Do ponto de vista do banco isso é ruim, já que você deixa de identificar clientes de risco.
- Falsos positivos = 0: O modelo nunca classifica alguém adimplente como inadimplente. Do ponto de vista do banco, isso evita perder "clientes bons", mas pode indicar que há algo de errado com ponto de corte, estando muito conservador.

Medidas de qualidade do modelo

Agora, vamos descobrir todas as medidas de qualidade do modelo que realizamos, começando pela acurácia.

Acurácia

A acurácia é a porcentagem de precisão do modelo. Ou seja, quantos valores do conjunto de teste o modelo preveu corretamente como adimplentes ou inadimplentes. O código abaixo exibe a acurácia do modelo:

```
acuracia = matriz$overall["Accuracy"]
cat("A acurácia do modelo é de:",round(acuracia,2)*100, "%.","\n")
```

```
## A acurácia do modelo é de: 85 %.
```

Como é possível observar, o modelo possui uma acurácia de 85%, decente, e próxima do ideal de 95%. Uma acurácia de 100% seria ruim, pois modelos com acurácia perfeita sofrem de *overfitting*, demonstrando que, talvez, ruídos do treinamento tenham prevalecido no modelo, reduzindo sua eficácia em situações reais de uso.

Sensibilidade

A sensibilidade mede a capacidade do modelo de identificar corretamente os casos positivos, ou seja, os casos em que `Loan Status` é igual a 1. O código abaixo exibe a sensibilidade:

```r
sensibilidade = matriz$byClass["Sensitivity"]
cat("A sensibilidade do modelo é de:", round(sensibilidade,2)*100, "%.","\n")
```

```
## A sensibilidade do modelo é de: 26 %.
```

A baixa sensibilidade obtida, de 26%, demonstra que muitos inadimplentes passam despercebidos pelo modelo. Mais uma vez, reforçamos a hipótese de que o ponto de corte de 50% não funciona bem para o modelo definido.

Especificidade

A especificidade mede a capacidade do modelo de identificar corretamente os casos negativos, ou seja, os casos em que `Loan Status` é igual a 0. O código abaixo exibe a especificidade:

```r
especificidade = matriz$byClass["Specificity"]
cat("A especificidade do modelo é de:", round(especificidade,2)*100,
"%.","\n")
```

```
## A especificidade do modelo é de: 100 %.
```

O modelo alcançou uma especificidade perfeita, de 100%, indicando que não há casos de falsos alarmes. Novamente, isso mostra um desequilíbrio no modelo com o ponto de corte de 50%, já que possuímos alta especificidade e baixa sensibilidade.

Precisão do modelo

A precisão do modelo mede a proporção de previsões positivas que estão corretas. O código abaixo exibe a precisão:

```r
precisao = matriz$byClass["Pos Pred Value"]
cat("A precisão do modelo é de:",round(precisao,2)*100,"%.","\n")
```

```
## A precisão do modelo é de: 100 %.
```

O modelo atingiu uma precisão de 100%¨, indicando que pode-se confiar totalmente quando o modelo diz que alguém é inadimplente. Ainda sim, com a baixa sensibilidade obtida, sabemos que o modelo é muito conservador, e a probabilidade de corte deve ser alterada.

Preditivo negativo

O preditivo negativo mede a proporção de previsões negativas que estão corretas. O código abaixo exibe o preditivo negativo:

```r
predneg = matriz$byClass["Neg Pred Value"]
cat("O valor preditivo negativo do modelo é de:", round(predneg,2)*100, "%.","\n")
```

```
## O valor preditivo negativo do modelo é de: 84 %.
```

O valor obtido para o preditivo negativo foi de 84%. Isso significa que 84% dos clientes que o modelo previu como adimplentes realmente são adimplentes. Ou seja, ainda existem inadimplentes que o modelo não consegue detectar.

Falsos positivos

Através do código abaixo, vamos obter o número de falsos positivos:

```r
falsopos = 1 - especificidade
cat("A taxa de falsos positivos é de:", round(falsopos,2)*100, "%.","\n")
```

```
## A taxa de falsos positivos é de: 0 %.
```

Falsos negativos

Em seguida, os falsos negativos:

```
falsoneg = 1 - sensibilidade
cat("A taxa de falsos negativos é de:", round(falsoneg,2)*100, "%.","\n")
```

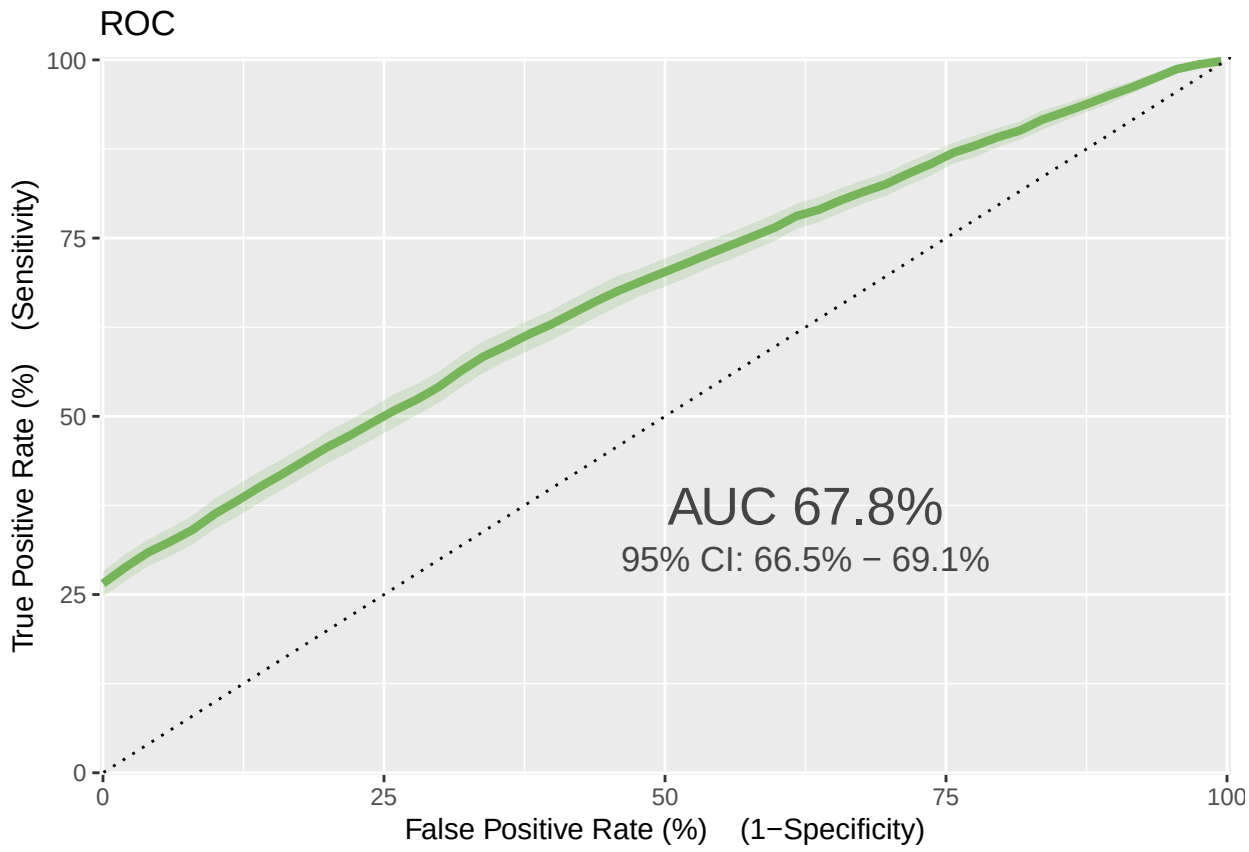
```
## A taxa de falsos negativos é de: 74 %.
```

Curva ROC

Como pudemos observar nas medidas de qualidade, o modelo está muito conservador. Então, vamos obter a curva ROC do modelo para encontrar o ponto de corte ideal a ser utilizado.

```
## [1] "Calculating AUC ..."
## [1] "(AUC) Sorting data ..."
## [1] "(AUC) Calculating ranks ..."
## [1] "AUC: 67.844761638732"
## [1] "Bootstrapping ROC curves"
## [1] "Eval AUC"
## [1] "Producing ROC plot"
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the classifierplots package.
##   Please report the issue at
##   <https://github.com/adefazio/classifierplots/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



A curva ROC está um pouco acima da linha do acaso, demonstrando que existe discriminação, mas não é muito forte. O modelo possui um AUC decente, de 67,8%, ou seja, ele possui uma discriminação relativamente aceitável.

A banda de confiança é estreita, indicando que o modelo é estável.

Agora, vamos obter exatamente o ponto de corte ótimo e sensibilidade e especificidade nesse ponto:

```
##   threshold sensitivity specificity
## 1 0.2028211   0.3079337    0.963308
```

Obtemos o ponto de corte ótimo em 0,203, ou 20,3%. Nesse ponto, a sensibilidade será de 0,308 (30,8%) e a especificidade 0,963 (96,3%).