Análise de Big Data via Machine Learning

LABDATA

Coordenação:

Prof. Dr. Adolpho Walter Pimazzi Canton

Profa. Dra. Alessandra de Ávila Montini

Machine Learning

Tema da Aula: Regressão Logística

Prof. Anderson França

Regressão Logística

A Regressão Logística foi desenvolvida pelo estatístico David Cox em 1958. É um **modelo de regressão onde a variável de resposta Y é categorica**.

A regressão nos permite **estimar a probabilidade de uma resposta** categórica com com base em uma ou mais variáveis preditoras (X). É possível dizer que a presença de um preditor aumenta (ou diminui) a probabilidade de um determinado resultado por uma porcentagem específica.

Análise de Regressão Logística

Tem o objetivo de projetar a probabilidade de ocorrer um evento de interesse.

Por exemplo projetar as probabilidades:

- Do cliente realizar pagamentos por internet banking;
- Do cliente realizar compras por internet banking;

Base de dados

Por hora, vamos abordar a logística no **caso do Y binário** - isto é, onde pode levar apenas dois valores, “0” e “1”, que representam resultados como passar/falhar, ganhar/perder, vivo/morto ou saudável/doente.

Vamos utilizar os seguintes pacotes:

```
library(tidyverse)   # dManipulação de dados e visualização
library(modelr)      # Fornece formas fáceis de implementar modelos e funções
library(broom)       # Ajuda a organizar as saídas dos modelos

library(ROCR)        # Curva ROC e AUC
```

Base de dados

Vamos utilizar os dados default fornecidos pelo pacote ISLR. Este é um conjunto de dados simulados que contém informações sobre dez mil clientes, como se é um cliente inadimplente, se é um estudante, qual o saldo médio do cliente e a renda do cliente.

```
#install.packages("ISLR")
(default <- as_tibble(ISLR::Default))
```

```
## # A tibble: 10,000 x 4
##    default student   balance    income
##     <fctr>  <fctr>     <dbl>     <dbl>
##  1      No      No  729.5265 44361.625
##  2      No     Yes  817.1804 12106.135
##  3      No      No 1073.5492 31767.139
##  4      No      No  529.2506 35704.494
##  5      No      No  785.6559 38463.496
##  6      No     Yes  919.5885  7491.559
##  7      No      No  825.5133 24905.227
##  8      No     Yes  808.6675 17600.451
##  9      No      No 1161.0579 37468.529
## 10      No      No    0.0000 29275.268
## # ... with 9,990 more rows
```

Regressão Logística

Por quê não utilizamos a **regressão linear** quando estamos trabalhando com resposta qualitativa?

Vamos supor que estamos tentando prever a condição médica de um paciente na sala de emergência com base em seus sintomas. Utilizando um exemplo simplificado, há três diagnósticos possíveis: *Acidente vascular cerebral (AVC)*, *overdose* e *ataque epiléptico*. Podemos considerar transformar esses valores em uma variável de resposta quantitativa, Y, da seguinte forma:

$$Y = \begin{cases} 1, & \text{Se AVC;} \\ 2, & \text{Se overdose;} \\ 3, & \text{Se ataque epiléptico.} \end{cases},$$

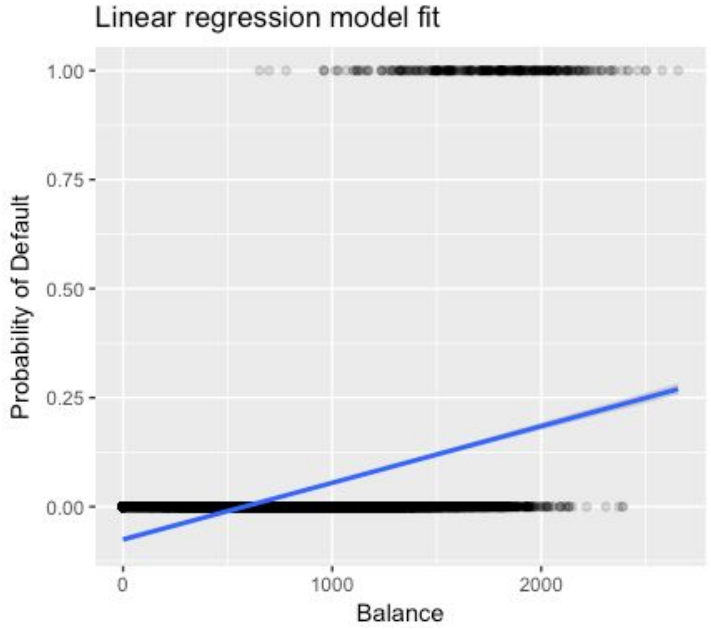
Regressão Logística

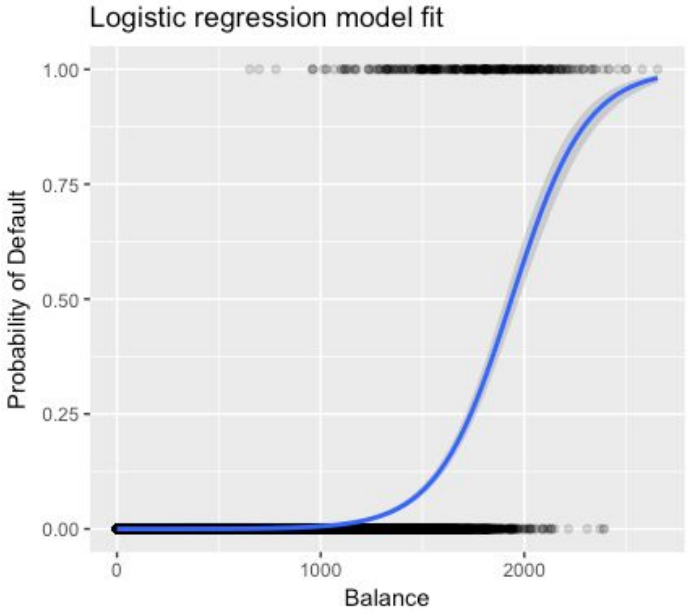
Utilizando esses códigos, os mínimos quadrados podem ser usados para ajustar um modelo de regressão linear para prever Y com base em um conjunto de preditores **X1,...,Xp**. Infelizmente essa codificação implica uma ordenação dos resultados, colocando overdose entre o AVC e implica que a diferença entre AVC e Overdose possuem a mesma diferença entre overdose e ataque epiléptico. Na prática, não há nenhum motivo particular para que isso seja necessário. Por exemplo, pode-se escolher uma codificação igualmente razoável

$$Y = \begin{cases} 1, & \text{if ataque epiléptico;} \\ 2, & \text{if AVC;} \\ 3, & \text{if Overdose.} \end{cases}$$

o que implicaria uma relação totalmente diferente entre as três condições. Cada uma dessas codificações produziria modelos lineares fundamentalmente diferentes que, em última instância, levariam a diferentes conjuntos de previsões em observações de teste.

Resultado do modelo





Para evitar este problema, devemos modelar **p(X)** usando uma função que fornece saídas entre 0 e 1 para todos os valores de X. Muitas funções atendem a esta descrição. Na regressão logística, usamos a função logística, que é definida na equação abaixo e ilustrado na figura direita acima

Exemplo Seguradora

Considere a variável de interesse Y como sendo uma variável aleatória com distribuição de probabilidade Bernoulli assumindo o valor Y=0 ou o valor Y=1.

No exemplo da seguradora, considere Y=0 para os clientes que não sofreram sinistro (não tiveram acidente com seu veículo) e Y=1 para os clientes que sofreram sinistro.

Y=1

Y=0

Exemplo Seguradora

No momento de venda de uma apólice de seguro de automóvel a seguradora precisa determinar a probabilidade de haver sinistro com o cliente.

Sejam:

p a probabilidade do cliente sofrer um sinistro (Y=1);

$$p = P(Y = 1)$$

1-p a probabilidade do cliente não sofrer um sinistro (Y=0);

$$1 - p = P(Y = 0)$$

Exemplo Seguradora

Suponha o exemplo em que pretende-se obter p=P(Y=1) considerando apenas a variável saldo na conta corrente (X).

Uma probabilidade é um valor entre 0 e 1.

$$0 \leq p \leq 1$$

A probabilidade *p* pode assumir um valor entre 0 e 1 e a variável X (saldo na conta corrente) **pode assumir qualquer valor** (positivo ou negativo).

Pode-se dizer que X pode variar do menos infinito ao mais infinito.

$$-\infty \leq X \leq +\infty$$

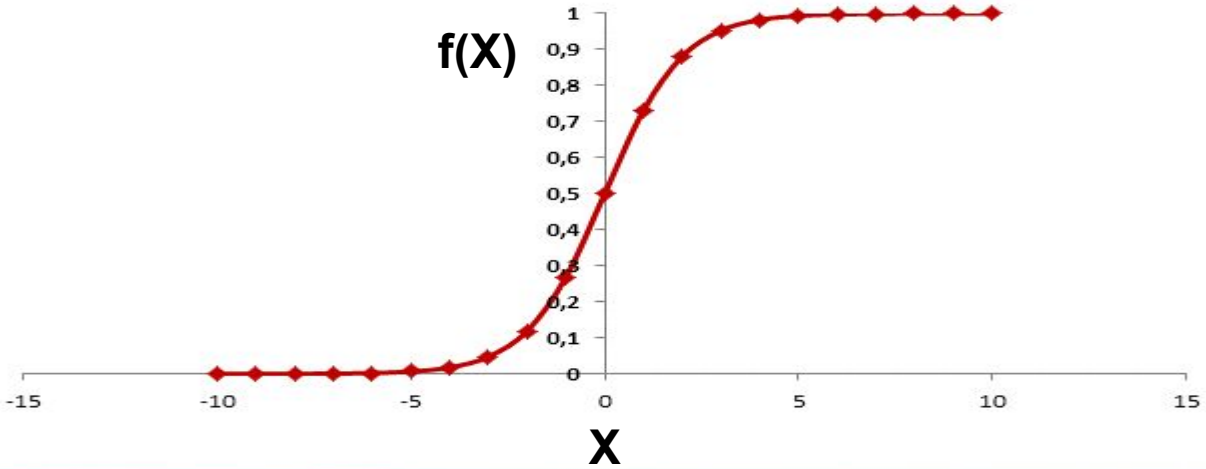
Função Logística

A **Função Logística** - f(x) - é uma função que assume valores entre 0 e 1 e a variável X (saldo na conta corrente) pode assumir qualquer valor.

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Função Logística

O gráfico representa a **Função Logística**. Os valores de X estão variando entre -10 e 10 e os valores de f(X) variando entre 0 e 1.



Exemplo Seguradora

No exemplo da seguradora, a probabilidade do cliente sofrer sinistro (*p*) pode ser obtida considerando várias variáveis como X_1= idade, X_2=sexo, X_3=valor do automóvel e X_4= tempo de habilitação.

Considerando que a função logística pode ser utilizada para obter a probabilidade do cliente sofrer sinistro (*p*) tem-se que:

$$p = P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4}}$$

Exemplo Seguradora

A probabilidade do cliente **sofrer sinistro** (p) também pode ser escrita como:

$$p = P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4)}}$$

Exemplo Seguradora

Considerando que a função logística pode ser utilizada para obter a probabilidade do **cliente não sofrer sinistro** (p) tem-se que:

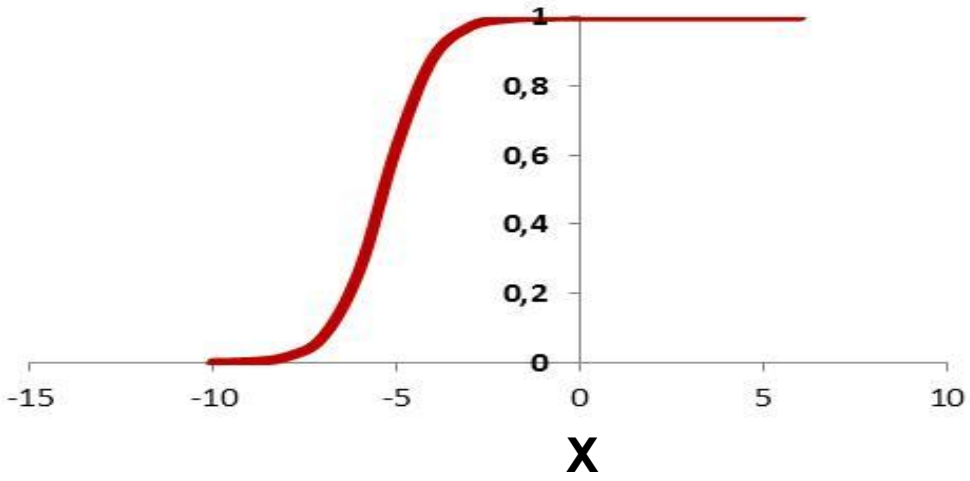
$$1-p=P(Y=0)=1-\frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}}{1+e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}}$$

A probabilidade 1-p também pode ser escrita como:

$$1-p=\frac{1}{1+e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}}$$

Função Logística

Quando o coeficiente da função logística β é positivo a probabilidade p cresce à medida que aumenta o valor de X. A figura apresenta a função logística com β=0,8.

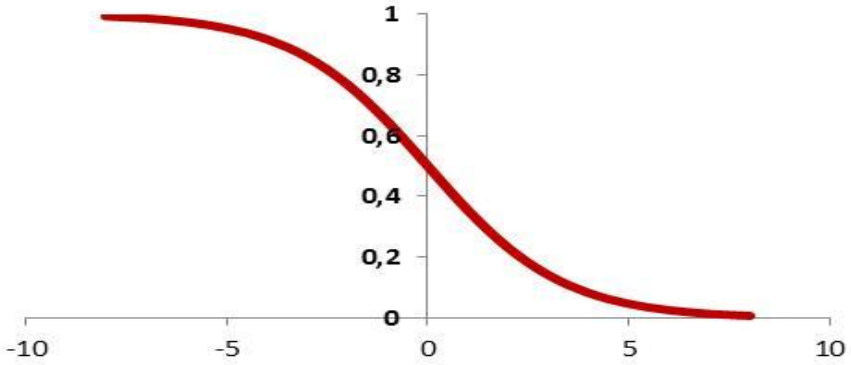


$$p = \frac{e^{\beta * X}}{1 + e^{\beta * X}}$$

$$p = \frac{e^{0,8 * X}}{1 + e^{0,8 * X}}$$

Função Logística

Quando o coeficiente da função logística β é negativo a probabilidade p decresce à medida que aumenta o valor de X. A figura apresenta a função logística com β=-0,6.



$$p = \frac{e^{-0,6*X}}{1 + e^{-0,6*X}}$$

Suposições

- O valor esperado do erro deve ser zero;
- Ausência de autocorrelação entre os erros;
- Ausência de correlação entre os erros e as variáveis independentes;
- Ausência de multicolinearidade entre as variáveis independentes.

Particionar os Dados

Vamos dividir nossa base em dois, 60% para a base de treinamento e 40% para teste, que será utilizada para avaliar o desempenho de nossos modelos em um conjunto dados fora da amostra.

```
#Fixar semente inicial
set.seed(123)

sample <- sample(c(TRUE, FALSE), nrow(default),
                 replace = T, prob = c(0.6,0.4))
train <- default[sample, ]
test <- default[!sample, ]
```

Regressão Logística Simples

Vamos ajustar um modelo de regressão logística para **prever a probabilidade de inadimplência de um cliente** com base no saldo médio do na conta.

A função glm ajusta modelos lineares generalizados, uma classe de modelos que inclui regressão logística. A sintaxe da função glm é semelhante à de lm, exceto que devemos passar o argumento family = binomial para que R possa executar uma regressão logística em vez de algum outro tipo de modelo linear generalizado.

```
model1 <- glm(default ~ balance, family = "binomial", data = train)
```

Por trás dessa função, o glm usa a máxima verossimilhança para ajustar o modelo.

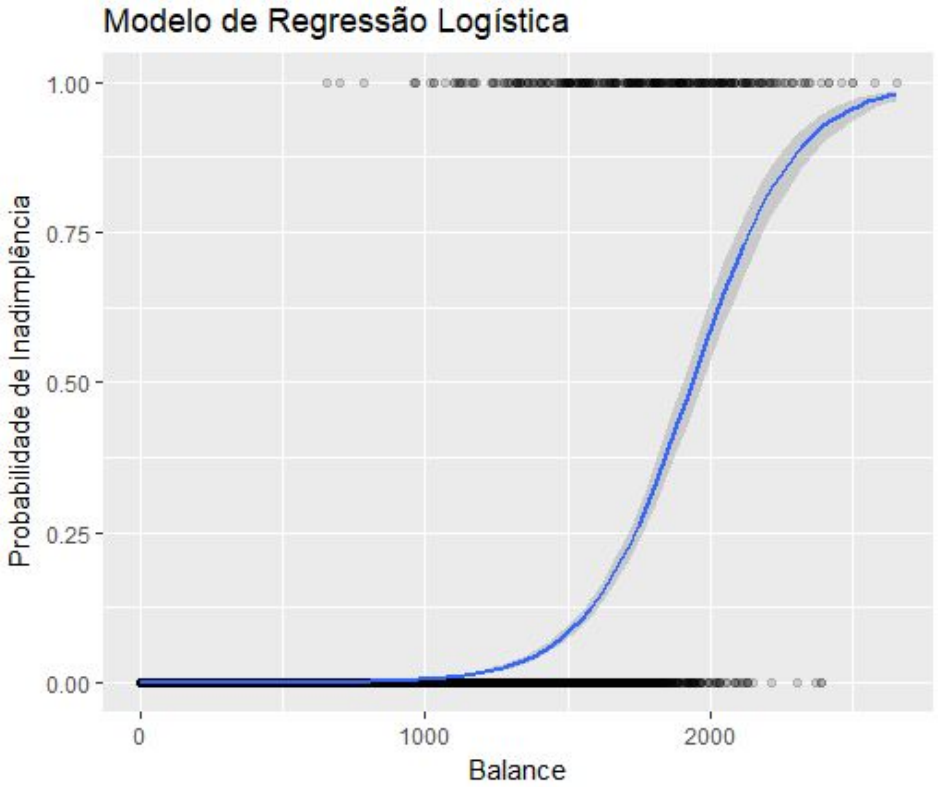
Regressão Logística Simples

A máxima verossimilhança é uma abordagem muito geral que é usada para ajustar a maioria dos modelos não-lineares. O que resulta é uma curva de probabilidade em forma de S que pode ser obtida utilizando o modelo abaixo

(observe que, para traçar a linha de ajuste de regressão logística, precisamos converter nossa variável de resposta para uma variável codificada binária [0,1]).

```
default %>%
  mutate(prob = ifelse(default == "Yes", 1, 0)) %>%
  ggplot(aes(balance, prob)) +
  geom_point(alpha = .15) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  ggtitle("Logistic regression model fit") +
  xlab("Balance") +
  ylab("Probability of Default")
```

Regressão Logística Simples



Regressão Logística Resumo do Modelo

Semelhante à regressão linear, podemos avaliar o modelo usando summary.

```
summary(model1)
```

```
Call:
glm(formula = default ~ balance, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2905  -0.1395  -0.0528  -0.0189   3.3346

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.101e+01  4.887e-01  -22.52   <2e-16 ***
balance      5.669e-03  2.949e-04   19.22   <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: 1723.03  on 6046  degrees of freedom
Residual deviance:  908.69  on 6045  degrees of freedom
AIC: 912.69

Number of Fisher Scoring iterations: 8
```

Regressão Logística Resumo do Modelo

```
Call:
glm(formula = default ~ balance, family = "binomial", data = tra

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2905  -0.1395  -0.0528  -0.0189   3.3346

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.101e+01  4.887e-01  -22.52   <2e-16 ***
balance      5.669e-03  2.949e-04   19.22   <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: 1723.03  on 6046  degrees of freedom
Residual deviance:  908.69  on 6045  degrees of freedom
AIC: 912.69

Number of Fisher Scoring iterations: 8
```

O desvio é análogo à soma dos cálculos em regressão linear e é uma medida da falta de ajuste para os dados em um modelo de regressão logística.

O **desvio nulo** representa a diferença entre um modelo com apenas o intercepto (que significa “sem preditores”) e um modelo saturado (um modelo com ajuste teoricamente perfeito).

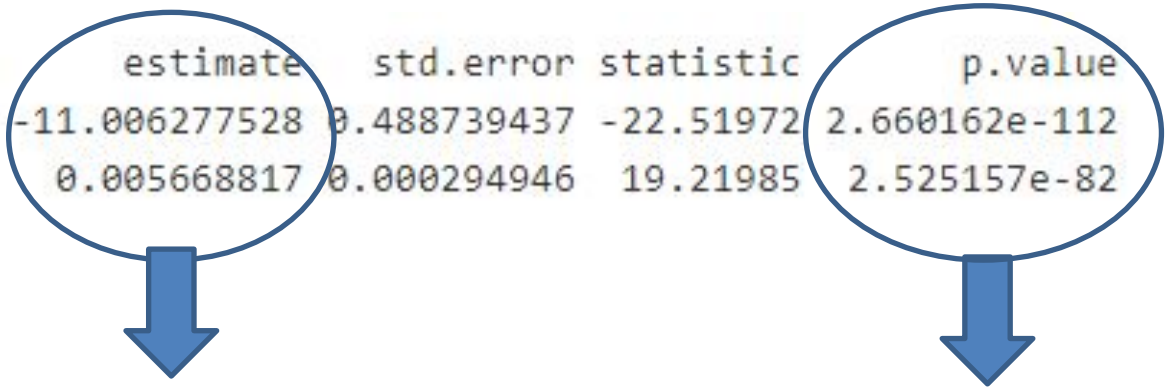
O objetivo é que o desvio do modelo (notado como desvio residual) seja menor, e valores menores indicam melhor ajuste. A este respeito, o modelo nulo fornece uma linha de base sobre a qual comparar dos modelos preditores.

Coeficientes do Modelo

Podemos obter as estimativas dos coeficientes e informações relacionadas que resultaram do ajuste do nosso modelo da seguinte forma:

```
tidy(model1)
```

```
##          term     estimate   std.error statistic      p.value
## 1 (Intercept) -11.006277528 0.488739437 -22.51972 2.660162e-112
## 2     balance   0.005668817 0.000294946  19.21985  2.525157e-82
```



Hipótese de interesse

Para verificar se a **variável balance** deve fazer parte do modelo deve-se testar a hipótese:

$$H_0 : \beta_1 = 0$$

$$H_1 : \beta_1 \neq 0$$

```
##          term      estimate   std.error statistic       p.value
## 1 (Intercept) -11.006277528 0.488739437 -22.51972 2.660162e-112
## 2     balance   0.005668817 0.000294946  19.21985  2.525157e-82
```

Como o nível descritivo (**Sig.=0,019**) < **0,10** rejeita-se a hipótese H_0, evidenciando que $\beta_1 \neq 0$, ou seja, a variável balance deve fazer parte do modelo.

Previsões

Uma vez que os coeficientes foram estimados, é simples calcular a probabilidade de inadimplência para qualquer saldo de cartão de crédito.

Usando as estimativas de coeficientes de nosso modelo, prevemos que a probabilidade de inadimplência para um indivíduo com um saldo de US $ 1.000 é inferior a 0,5%

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-11.0063+0.0057\times 1000}}{1+e^{-11.0063+0.0057\times 1000}} = 0.004785$$

Previsões

Podemos prever a probabilidade de inadimplência usando a função predict (certifique-se de incluir type = “response”). Aqui, vamos comparar a probabilidade de inadimplência com base em saldos de USS 1000 e USS 2000.

```
predict(model1, data.frame(balance = c(1000, 2000)),
        type = "response")
```

```
##           1           2
## 0.004785057 0.582089269
```

Nesse modelo, podemos notar que quando o saldo se move de USS 1000 para USS 2000, a probabilidade de inadimplência aumenta significativamente, de 0,5% para 58%!

Previsões

Podemos também usar preditores qualitativos em nosso modelo de regressão logística. Como exemplo, podemos incluir a variável **student.**

```
model2 <- glm(default ~ student, family = "binomial", data = train)
tidy(model2)
```

```
##          term   estimate  std.error statistic     p.value
## 1 (Intercept) -3.5534091 0.09336545 -38.05914 0.000000000
## 2  studentYes  0.4413379 0.14927208   2.95660 0.003110511
```

Isso indica que os alunos tendem a ter maiores probabilidades de inadimplência do que não estudantes.

De fato, esse modelo sugere que um aluno tenha quase o dobro das chances de inadimplência do que não estudantes.

Previsões

Podemos prever a probabilidade de inadimplência usando a função predict (certifique-se de incluir type = “response”). Aqui, vamos comparar a probabilidade de inadimplência com base se o indivíduo é estudante ou não

```
predict(model2, data.frame(student = factor(c("Yes", "No"))),
        type = "response")
```

```
##           1           2
## 0.04261206 0.02783019
```

Regressão Logística Múltipla

Nós também podemos extender nosso modelo como visto na equação da logístca de forma que podemos prever uma resposta binária utilizando multiplos preditores **X=(X1,...,Xp)X=(X1,...,Xp)** onde os *p* preditores são:

$$p(X) = \frac{e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}$$

Regressão Logística Múltipla

Nós também podemos extender nosso modelo como visto na equação da logístca de forma que podemos prever uma resposta binária utilizando multiplos preditores **X=(X1,...,Xp)X=(X1,...,Xp)** onde os *p* preditores são:

$$p(X) = \frac{e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X+\cdots+\beta_p X_p}}$$

Regressão Logística Múltipla

Vamos ajustar um modelo que prevê a probabilidade de inadimplência com base no saldo (balance), na renda (income) (em milhares de dólares) e nas variáveis de status do aluno(student).

```
model3 <- glm(default ~ balance + income + student, family = "binomial",
              data = train)
tidy(model3)
```

```
##          term      estimate    std.error   statistic      p.value
## 1 (Intercept) -1.090704e+01 6.480739e-01 -16.8299277 1.472817e-63
## 2     balance  5.907134e-03 3.102425e-04  19.0403764 7.895817e-81
## 3      income -5.012701e-06 1.078617e-05  -0.4647343 6.421217e-01
## 4  studentYes -8.094789e-01 3.133150e-01  -2.5835947 9.777661e-03
```

Previsões

Então podemos facilmente fazer previsões utilizando esse modelo. Por exemplo, um aluno com saldo de cartão de crédito de USS 1.500 e uma receita de USS 40.000 tem uma probabilidade estimada de inadimplência de

$$\hat{p}(X) = \frac{e^{-10.907+0.00591\times 1{,}500-0.00001\times 40-0.809\times 1}}{1+e^{-10.907+0.00591\times 1{,}500-0.00001\times 40-0.809\times 1}} = 0.054$$

Um não estudante com o mesmo saldo e renda tem uma probabilidade estimada de não cumprimento de

$$\hat{p}(X) = \frac{e^{-10.907+0.00591\times 1{,}500-0.00001\times 40-0.809\times 0}}{1+e^{-10.907+0.00591\times 1{,}500-0.00001\times 40-0.809\times 0}} = 0.114$$

Regressão Logística Múltipla

Vamos ajustar um modelo que prevê a probabilidade de inadimplência com base em nosso modelo

```
new.df <- tibble(balance = 1500, income = 40, student = c("Yes", "No"))
predict(model3, new.df, type = "response")
```

```
##          1          2
## 0.05437124 0.11440288
```

Assim, vemos isso para o saldo e a renda (embora a renda seja insignificante), um aluno tem cerca da metade da probabilidade de inadimplência do que um não estudante.

Medida de Ajuste

Pseudo R^2:

A métrica de pseudo R^2 mais utilizado é o McFadden's R^2, que é definido como:

$$1 - \frac{ln(LM_1)}{ln(LM_0)}$$

Onde ln(LM1) é o valor de probabilidade de log para o modelo ajustado e ln(LM0) é a probabilidade do log para o modelo nulo com apenas um intercepto como preditor.

```
list(model1 = pscl::pR2(model1)["McFadden"],
     model2 = pscl::pR2(model2)["McFadden"],
     model3 = pscl::pR2(model3)["McFadden"])
```

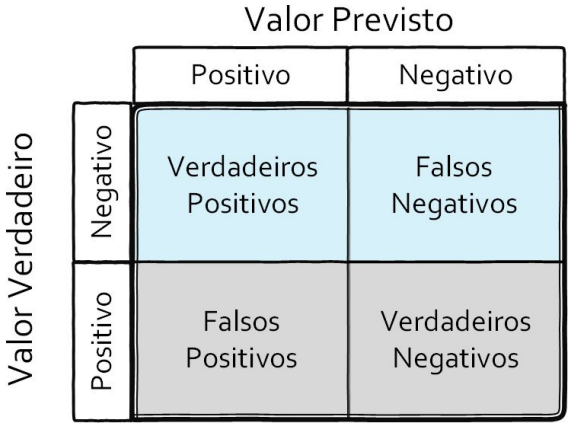
Taxa de Classificação

Quando criamos nosssos modelos de previsão, a métrica mais crítica diz respeito sobre o quão bem o modelo faz as previsões da variável target em uma nova base de dados. Primeiro, precisamos usar os modelos estimados para prever valores em nosso conjunto de dados de treinamento (train).

```
test.predicted.m1 <- predict(model1, newdata = test, type = "response")
test.predicted.m2 <- predict(model2, newdata = test, type = "response")
test.predicted.m3 <- predict(model3, newdata = test, type = "response")
```

```
list(
  model1 = table(test$default, test.predicted.m1 > 0.5) %>% prop.table() %>% round(3),
  model2 = table(test$default, test.predicted.m2 > 0.5) %>% prop.table() %>% round(3),
  model3 = table(test$default, test.predicted.m3 > 0.5) %>% prop.table() %>% round(3)
)
```

Matriz de Confusão



- **Positivos verdadeiros**: são casos em que previmos que o cliente seria inadimplente e de fato o fizeram.

- **Negativos verdadeiros**: nós previmos a adimplência, e o cliente não virou inadimplência.

- **falsos positivos** : previámos sim, mas na verdade não foram inadimplentes (Também conhecido como um “erro de Tipo I.”)

- **falsos negativos**: nós previamos não, mas eles se tornaram inadimplentes. (Também conhecido como “erro tipo II”).

Análise

Os resultados mostram que model1 e model3 são muito semelhantes. 96% das observações previstas são verdadeiras negativas e cerca de 1% são verdadeiras positivas.

Ambos os modelos têm um erro de tipo II inferior a 3%, no qual o modelo prevê que o cliente não será inadimplentes, mas eles realmente o fizeram.

E ambos os modelos têm um erro de tipo I de menos de 1% em que os modelos prevêem que o cliente será padrão, mas nunca o fizeram.

Os resultados do modelo 2 são notavelmente diferentes; Este modelo prevê com precisão os não-inadimplentes (resultado de 97% dos dados serem não-inadimplentes), mas nunca prevê os clientes que são inadimplentes!

```
## $model1
##
##       FALSE  TRUE
##   No  0.962 0.003
##   Yes 0.025 0.010
```

```
## $model2
##
##       FALSE
##   No  0.965
##   Yes 0.035
```

```
## $model3
##
##       FALSE  TRUE
##   No  0.963 0.003
##   Yes 0.026 0.009
```

Tabela de Classificação

Nós também queremos entender as taxas de missclassification (aka error). Não vemos muita melhoria entre os modelos 1 e 3 e, embora o modelo 2 tenha uma taxa de erro baixa, não esqueça que isso nunca prevê com precisão os clientes que realmente são inadimplentes.

```
test %>%
  mutate(m1.pred = ifelse(test.predicted.m1 > 0.5, "Yes", "No"),
         m2.pred = ifelse(test.predicted.m2 > 0.5, "Yes", "No"),
         m3.pred = ifelse(test.predicted.m3 > 0.5, "Yes", "No")) %>%
  summarise(m1.error = mean(default != m1.pred),
            m2.error = mean(default != m2.pred),
            m3.error = mean(default != m3.pred))
```

Outras métricas

Com os modelos de classificação, vamos ouvir falar bastante dos termos sensibilidade e especificidade ao caracterizar o desempenho do modelo. A sensibilidade é sinônimo de precisão

Sensibilidade (acertos)

P(classificar um cliente como inadimplente / que ele é inadimplente)

Especificidade (acertos)

P(classificar um cliente como adimplente/ que ele é adimplente)

Erro de classificação

1 – Especificidade=P(classificar um cliente como inadimplente / que ele é adimplente)

Tabela de classificação[a]

Observado			Previsto		
			ST		Porcentagem correta
			0	1	
Etapa 1	ST	0	45	6	88,2
		1	4	37	90,2
	Porcentagem global				89,1

a. O valor de corte é ,500

Sensibilidade (acertos)

P(classificar um cliente como inadimplente/ que ele é inadimplente)

P(PREVISTO = 1 / OBSERVADO = 1)=37/41=0,9024

Especificidade (acertos)

P(classificar um cliente como adimplente/ que ele é adimplente)

P(PREVISTO = 0 / OBSERVADO = 0)=45/51=0,88

Tabela de classificação[a]

Observado			Previsto		
			ST		Porcentagem correta
			0	1	
Etapa 1	ST	0	45	6	88,2
		1	4	37	90,2
	Porcentagem global				89,1

a. O valor de corte é ,500

Erro de classificação

1 – Especificidade=P(classificar um cliente como inadimplente / que ele é adimplente)

1 – Especificidade = 1-0,88 =0,12

ROC

Pela análise da curva ROC, escolhemos o ponto de corte referente a combinação da sensibilidade e 1-especificidade que mais se aproxima do canto superior esquerdo do gráfico.

Curva ROC (Receiver Operating Characteristic)

- Para elaborar o gráfico é necessário variar o ponto de corte de 0 a 1
- A área total do gráfico = 1

Área sob a Curva

- Quando a área sob a curva = 0,5 o modelo não tem poder de discriminação
- Quando a 0,5 < área < 0,7 o modelo possui discriminação fraca
- Quando a 0,71 < área < 0,8 o modelo possui discriminação aceitável
- Quando a 0,81 < área < 0,9 o modelo possui discriminação boa
- Quando a área > 0,9 o modelo possui discriminação ótima

ROC

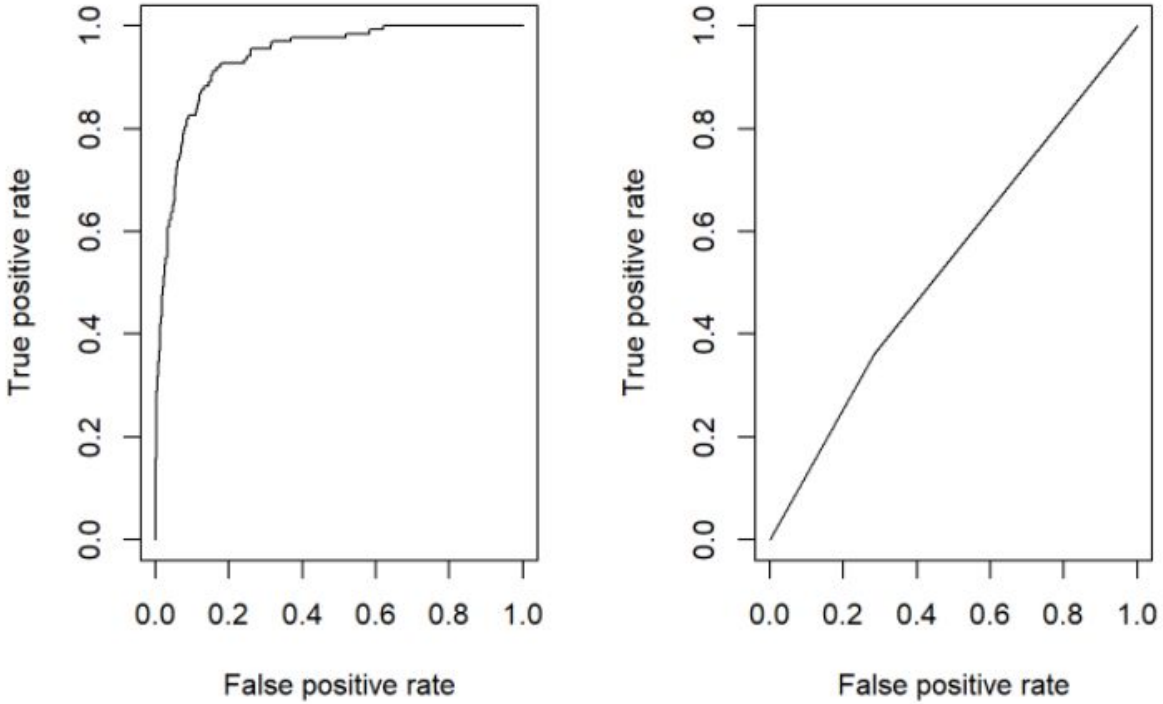
```
library(ROCR)

par(mfrow=c(1, 2))

prediction(test.predicted.m1, test$default) %>%
  performance(measure = "tpr", x.measure = "fpr") %>%
  plot()

prediction(test.predicted.m2, test$default) %>%
  performance(measure = "tpr", x.measure = "fpr") %>%
  plot()
```

ROC


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

AUC

E para computar a AUC numericamente, podemos usar o código abaixo. Lembre-se, AUC irá variar de .50 - 1.00. Assim, o modelo 2 é um modelo de classificação muito fraco, enquanto o modelo 1 é um modelo de classificação muito bom.

```
# modelo 1 AUC
prediction(test.predicted.m1, test$default) %>%
  performance(measure = "auc") %>%
  .@y.values
```

```
# modelo 2 AUC
prediction(test.predicted.m2, test$default) %>%
  performance(measure = "auc") %>%
  .@y.values
```

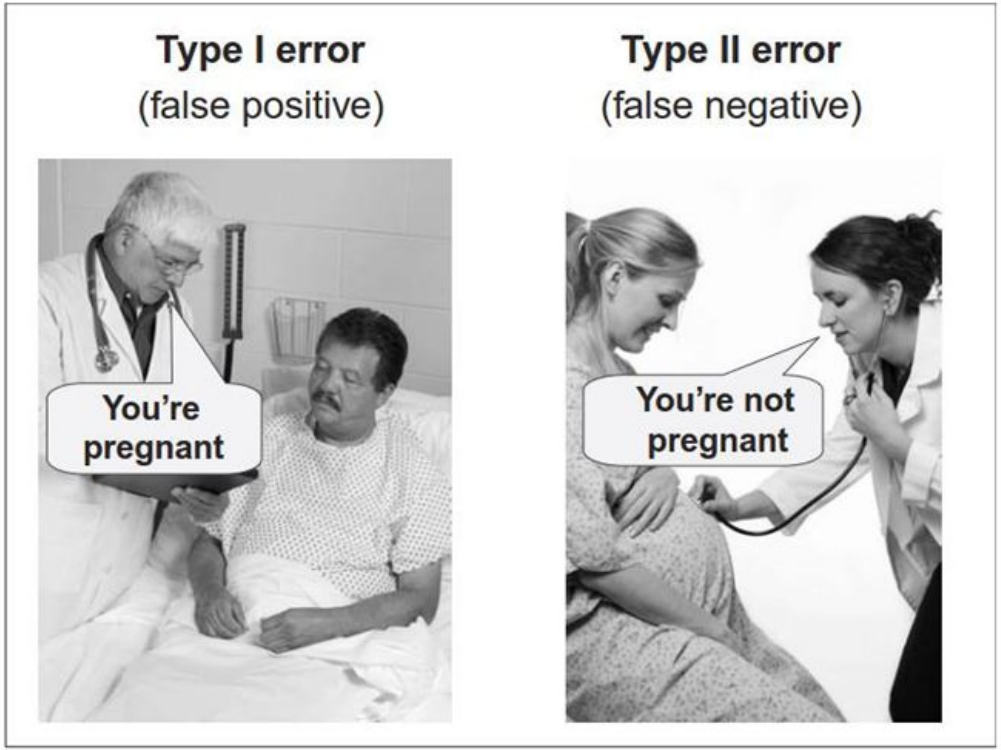


Figure 3.1 Type I and Type II errors

Referência Bibliográfica

Exemplos extraídos de: UC Business Analytics R Programming Guide, 2018
Cox, D. R.; SNELL, E. J. Analysis of binary data. 2. ed. London: Champman and Hall, 1989

Hosmer, David W.; Lemeshow, Stanley. Applied logistic regression. New York: Wiley, 1989

Johnson, Richard A.; Wichern, Dean W. Applied multivariate statistical analysis. New Jersey: Prentice Hall, 1998

Menard, Scott W. Applied logistic regression analysis. Thousands Oaks, Calif: Sage Publications, n. 7, 1995

Gordon, S. Linoff; Berry, M. J. A. Data Mining Techniques : For Marketing, Sales and Custmer Relationship Management. Third Edition. Wiley, 2011