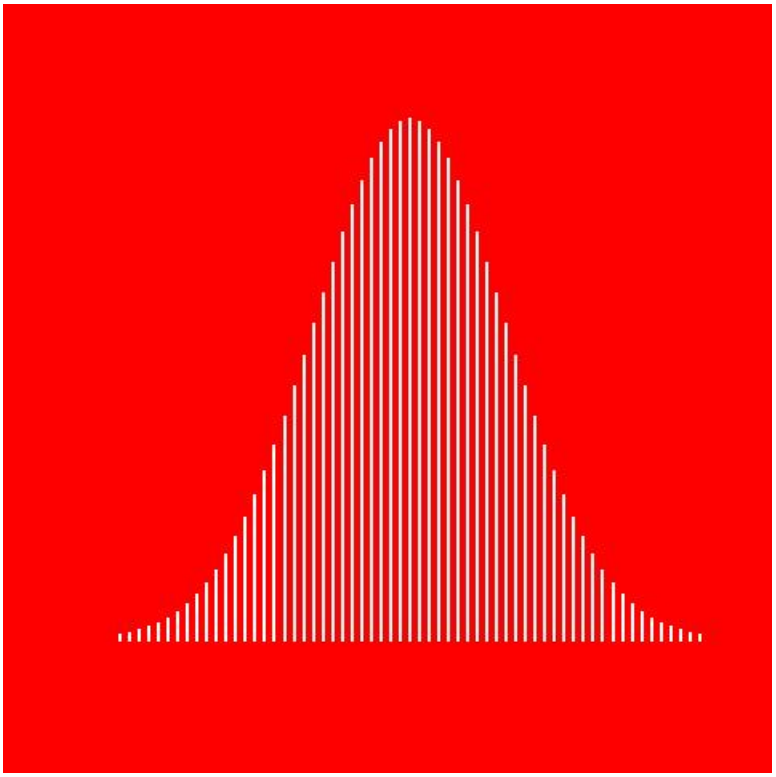
Machine Learning II
Regressão Logistica



Brasília - DF
janeiro de 2019

Sumário

Regressão Logística - Introdução

O modelo de regressão linear pressupõe que a variável resposta Y é quantitativa. Mas em muitas situações, a variável resposta é qualitativa. Por exemplo, a cor dos olhos é qualitativa, que pode assumir os valores azul, marrom ou verde.

Prever uma resposta qualitativa para classificar uma observação pode ser referida com tarefa de classificação, uma vez que envolve atribuir a observação a uma categoria ou classe. Por outro lado, muitas vezes os métodos para classificação, primeiro prever a probabilidade da observação pertencer a cada uma das categorias, como base para fazer a classificação. Nesse sentido, eles também se comportam como métodos de regressão.

Problema de Classificação

A tarefa de classificação associa ou classifica um objeto em uma das classes presentes na análise, também busca prever uma classe de um novo objeto automaticamente. Por exemplo, uma base de dados que armazena características de clientes, baseando em históricos de transações anteriores, podem-se classificar estes clientes em categorias para liberação de crédito. Um novo cliente poderá ser classificado em uma das categorias definidas, de acordo com suas características. Problemas de classificação ocorrem frequentemente e incluem:

1. Uma pessoa chega na sala de emergência com um conjunto de sintomas que possivelmente poderia ser atribuído a uma das três condições médicas.

2. Um serviço bancário online deve poder determinar se deve ou não uma transação que está sendo realizada no site é fraudulenta, com base do endereço IP do usuário, histórico de transações anteriores e assim por diante.

3. Com base nos dados da sequência de DNA para um número de pacientes com e sem uma dada doença, um biólogo gostaria de descobrir qual das mutações no DNA são deletérias (causadoras de doenças) e não são.

Em uma tarefa de classificação, há um conjunto de observações de treinamento $(x_1, y_1), \cdots, (x_n, y_n)$ que podemos usar para construir um classificador. Queremos que nosso classificador tenha um bom desempenho não apenas no treinamento dados, mas também em observações de teste que não foram usadas para treinar o classificador.

Vamos ilustrar o conceito de classificação usando data frame simulado `Default`. Estamos interessados em prever se um o indivíduo será inadimplente em seu pagamento com cartão de crédito, com base em renda anual (`income`) e saldo mensal (`balence`) do cartão de crédito. O conjunto de dados é exibido na Figura 1. Nós graficamos `income` e `balence` para um subconjunto de 10.000 indivíduos. O painel esquerdo da Figura 1 exibe indivíduos que entraram em default em um determinado mês em laranja, e aqueles quem não entrarão em azul. A taxa geral de inadimplência é de cerca de 3%, e que indivíduos os inadimplentes tendem a ter saldos de cartão de crédito mais altos do que aqueles que não o entraram . No painel à direita da Figura 1, dois pares de boxplots são exibidos. O primeiro mostra os valores observados em `balance` para cada valor da variável binária `default`; o segundo é painel semelhante para a variável `income`. O objetivo é construir um modelo para prever o `default` (Y) para qualquer valor dado de `balance` (X1) e `income` (X2). Como Y não é quantitativo, o modelo de regressão linear simples não é apropriado.

Vale a pena notar que a Figura 1 exibe uma relação muito pronunciada entre o `balance` e a resposta `default`. Na maioria das aplicações reais, a relação entre o preditor e a resposta não será tão forte. No entanto, para ilustrar os procedimentos de classificação discutidos aqui, usamos um exemplo em que a relação entre o preditor e a resposta é um pouco exagerada.

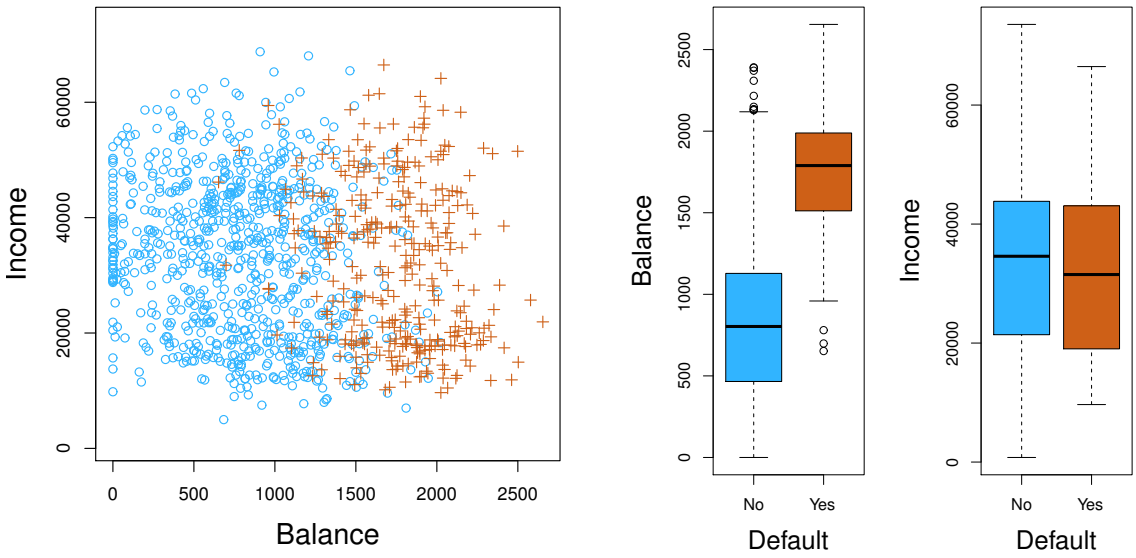


Figura 1: O conjunto de dados `Default`. Esquerda:`income` - os rendimentos anuais - e `balance` - saldos mensais de cartão de crédito - de um número de indivíduos. Os indivíduos que entraram em `default` seus pagamentos com cartão de crédito são mostrados em laranja, e aqueles que não foram mostrados em azul. Centro: Boxplots de *balence* em função do `default`. Direita: Boxplots de `income` em função de `default`.

Regressão Logística

Considere novamente o data frame `Default`, no qual a variável `default` é uma das duas categorias, `Yes` ou `No`. Em vez de modelar essa resposta, Y diretamente, a regressão logística modela a probabilidade de Y pertencer a uma dessas categorias.

Para os dados `Default`, a regressão logística modela a probabilidade de default. Por exemplo, a probabilidade de default dado saldo pode ser escrito como

$$Pr(default = Yes|balance).$$

Os valores de $Pr(default = Yes|balance)$, que abreviamos $p(balance)$, irá variar entre 0 e 1. Então, para qualquer valor de `balance`, uma previsão pode ser feita para `default`. Por exemplo, pode-se prever $default = Yes$ para qualquer indivíduo para quem $P(balance) > 0,5$. Alternativamente, se uma empresa deseja ser conservadora na previsão de indivíduos que correm risco de inadimplência, então eles podem escolher usar um limite mais baixo, como $p(balance) > 0,1$.

Modelo Logístico

Como devemos modelar a relação entre $p(x) = pr(y = 1|x)$ e X (Por conveniência, estamos usando a codificação 0/1 genérica para a resposta). Um modelo de regressão linear para representar estas probabilidades:

$$p(X) = p_0 + p_1 X \tag{1}$$

Se usarmos essa abordagem para prever o `default =Yes` usando o `balance`, então obtemos o modelo mostrado no painel esquerdo da Figura 2. Aqui vemos o problema com esta abordagem: para `balance` próximos de zero, predizemos probabilidade negativa de inadimplência; se tivéssemos que prever para

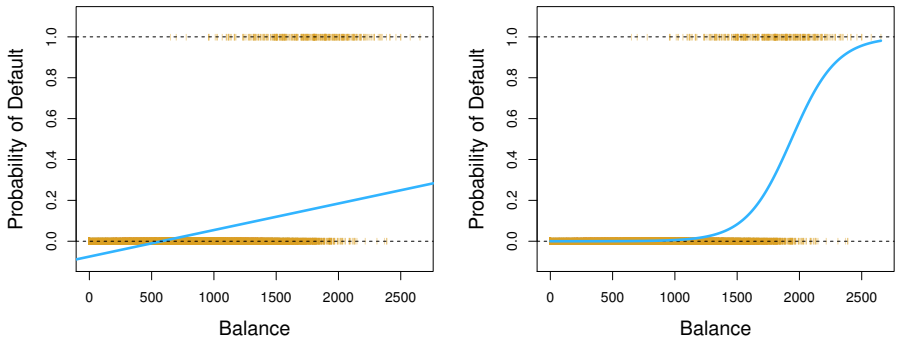


Figura 2: Classificação usando os dados `Default`. Esquerda: probabilidade estimada de `default` usando regressão linear. Algumas probabilidades estimadas são negativas! Os pontos laranja indicam os valores 0/1 codificados por padrão (No ou Yes). Direita: Probabilidades previstas de default usando a regressão logística. Todas as probabilidades estão entre 0 e 1.

saldos muito grandes, obteríamos valores maiores que 1. Essas previsões não são sensatas, uma vez que a verdadeira probabilidade de inadimplência, independentemente do saldo do cartão de crédito, deve estar entre 0 e 1. Esse problema não é exclusivo dos dados de inadimplência para dados de crédito. Sempre que uma linha reta é ajustada a uma resposta binária codificada como 0 ou 1, em princípio, podemos sempre prever $p(X) < 0$ para alguns valores de X e $p(X) > 1$ para outros (a menos que o intervalo de X seja limitado).

Para evitar esse problema, devemos modelar $p(X)$ usando uma função que fornece saídas entre 0 e 1 para todos os valores de X. Na regressão logística, usamos a *função logística*,

$$P(X) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 X}} \tag{2}$$

Para ajustar o modelo (2), usamos um método chamado máxima verossimilhança, que apresentaremos a seguir. O painel à direita da Figura 2 ilustra a probabilidade o ajuste do modelo de regressão logística aos dados `Default`. Observe que para `balance` baixos agora prever a probabilidade de inadimplência tão perto, mas nunca abaixo de zero. Da mesma forma, para `balance` elevados, prevemos uma probabilidade de inadimplência perto, mas nunca acima, um. A função logística sempre produzirá uma curva em forma de S desta forma, e assim, independentemente do valor de X, obterá uma previsão sensata. Também vemos que o modelo logístico é capaz de capturar melhor o intervalo de probabilidades do que o modelo de regressão linear, gráfico da esquerda. A probabilidade ajustada média em ambos os casos é 0,0333 (média dos dados de treinamento), que é o mesmo que proporção de inadimplentes no conjunto de dados. Depois de um pouco de manipulação de (2), descobrimos que

$$\frac{P(x)}{1 - p(x)} = e^{\beta_0+\beta_1 X} \tag{3}$$

A quantidade $p(X)/(1-p(X))$ é chamada de `odds` e pode assumir qualquer valor entre 0 e ∞. Valores `odds` próximos de 0 e ∞ de indicam valores muito baibabildeidades proxas ou probabilidades muito altas de `default`, respectivamente. Por exemplo, em média 1 em cada 5 pessoas com `odds` de 1/4 será o `default`, já que $p(X) = 0.2$ implica um `odds` de $\frac{0,2}{1-0,2} = 1/4$. Da mesma forma, em média, nove em cada dez pessoas com uma probabilidade de 9 será o `default`, já que p $(X) = 0,9$ implica em uma `odds` de $\frac{0,9}{1-0,9} = 9$. Os `odds` são tradicionalmente usadas em vez de probabilidades em corridas de cavalos, uma vez que eles se relacionam mais naturalmente com a estratégia correta de apostas. Tomando o logaritmo de ambos os lados de (3), chegamos a

	Coefficient	Std. erro	Z-statistic	P-value
intercept	-10.6513	0.3612	-29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Tabela 1: Para os dados `Default`, coeficientes estimados da regressão logística modelo que prevê a probabilidade de inadimplência usando o saldo. Uma unidade O aumento do saldo está associado a um aumento nas probabilidades de0,0055 unidades.

$$\log\left(\frac{P(x)}{1-p(x)} = e^{\beta_0+\beta_1 x}\right) = \beta_0 + \beta_1 X \tag{4}$$

O lado esquerdo é chamado *log-odds* ou *logit*. Nós vemos que as O modelo de regressão logit (2) tem o *logit* que é linear em X.

Em um modelo de regressão linear, β_1 é a variação média em Y associada a um aumento de uma unidade em X. Em contraste, em um modelo de regressão logística, aumentando X por uma unidade muda as probabilidades de log por β_1 (4), ou equivalentemente multiplica as probabilidades por e^{β_1} (3). Contudo, porque a relação entre $p(X)$ e X em (2) não é uma linha reta, β_1 não corresponde à alteração em $p(X)$ associada a uma unidade aumento em X. A quantidade que $p(X)$ muda devido a uma mudança de uma unidade em X dependerá do valor atual de X. Mas, independentemente do valor de X, se β_1 for positivo, o aumento de X será associado ao aumento de $p(X)$, e se β_1 é negativo, então o aumento de X será associado à diminuição $p(X)$. O fato de que não existe uma relação linear entre $p(X)$ e X, e o fato de que a taxa de mudança em $p(X)$ por unidade muda em X depende do valor atual de X, também pode ser visto pela inspeção do painel direito da Figura 2.

Estimação dos Coeficientes

Os coeficientes β_0 e β_1 em (2) são desconhecidos e devem ser estimados com base nos dados de treinamento disponíveis. Em uma regressão linear, usamos a abordagem dos mínimos quadrados para estimar os coeficientes desconhecidos. Apesar de podermos usar mínimos quadrados (não lineares) para ajustar o modelo (4), mais método geral de máxima verossimilhança é o preferido, pois possui melhores propriedades estatísticas. A intuição básica por trás do uso da máxima verossimilhança para ajustar um modelo de regressão logística é a seguinte: buscamos estimativas para β_0 e β_1 tal que a probabilidade prevista $\hat{p}(x_i)$ de default para cada indivíduo, usando (2), corresponde, tanto quanto possível, ao observado pelo indivíduo status padrão. Em outras palavras, tentamos encontrar $\hat{\beta}_0$ e $\hat{\beta}_1$ de tal forma que o estas estimativas para o modelo para $p(X)$, dado em (2), produz um número perto de um para todos os indivíduos que faltaram, e um número próximo de zero para todos os indivíduos que não o fizeram. Esta intuição pode ser formalizada usando um equação matemática chamada de função de verossimilhança:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}} (1 - p(x_i')) \tag{5}$$

As estimativas β_0 e β_1 são escolhidas para maximizar essa função de verossimilhança.

Máxima verossimilhança é uma abordagem muito geral que é usada para dos modelos não lineares. No ajuste de regressão linear, a abordagem dos mínimos quadrados é de fato um caso especial de máxima verossimilhança. No entanto, em geral, a regressão logística e outros modelos podem ser facilmente ajustados usando um pacote de software estatístico como R, e por isso não precisamos nos preocupar com os detalhes do procedimento de ajuste de máxima verossimilhança.

A Tabela 1 mostra as estimativas dos coeficientes e informações relacionadas resultado da montagem de um modelo de regressão logística nos dados `Default`, a fim para prever a probabilidade de `default = Yes` usando `balance`. Nós vemos que $\beta_1 = 0,0055$; isso indica que um aumento no equilíbrio está associado a um aumento na probabilidade de inadimplência. Para ser preciso, um aumento de uma

unidade em `balance` está associado com um aumento nas probabilidades de log de padrão de 0,0055 unidades.

Muitos aspectos da saída da regressão logística mostrados na Tabela 1 são semelhante à saída de regressão linear. Por exemplo, podemos medir a precisão das estimativas dos coeficientes calculando seus erros padrão. A *estatística-z* na Tabela 1 desempenha o mesmo papel que a *estatística-t* na saída de regressão linear. Por exemplo, a *estatística-z* associada a β_1 é igual a $\frac{\beta_1}{SE(\hat{\beta}_1)}$, e assim valor grande (absoluto) da *estatística-z* indica evidência contra o valor nulo hipótese $H_0 : \beta_1 = 0$. Essa hipótese nula implica que $p(X) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$ em outras palavras, que a probabilidade de inadimplência não depende do equilíbrio. Como o valor p associado ao equilíbrio na Tabela 1 é pequeno, podemos rejeitar H_0. Em outras palavras, concluímos que existe de fato uma associação entre equilíbrio e probabilidade de inadimplência. Interceptação estimada na Tabela 1 normalmente não é de interesse; seu principal objetivo é ajustar a média ajustada probabilidades à proporção de uns nos dados.

Predição

Uma vez que os coeficientes foram estimados, é uma questão simples calcular a probabilidade de inadimplência para qualquer saldo de cartão de crédito. Por exemplo, usando as estimativas dos coeficientes fornecidas na Tabela 1, prevemos que o padrão probabilidade de um indivíduo com um saldo de $ 1,000 é

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1}}{1+e^{-10.6513+0.0055\times 1}} \tag{6}$$

que é inferior a 1%. Em contraste, a probabilidade prevista de inadimplência para um indivíduo com `balance` de U$ 2,000 é muito maior, e é igual a 0,586 ou 58,6%.

Pode-se usar preditores qualitativos com o modelo de regressão logística usando a abordagem da variável `dummy`. Como um exemplo, o conjunto de dados `default` contém a variável qualitativa `student`. Para encaixar o modelo, basta criar uma variável `dummy` que assume um valor de 1 para estudante e 0 para o não estudantes. O modelo de regressão logística prever um aumento de probabilidade de inadimplência para o estudante, como pode ser visto na Tabela 2. O coeficiente associado à variável `dummy` é positivo,

	Coefficient	Std. erro	Z-statistic	P-value
intercept	-3.5041	0.0707	-49.55	<0.0001
student	0.4049	0.1150	3.52	0.0004

Tabela 2: Para os dados `Default`, os coeficientes estimados para o modelo de regressão logística prevê a probabilidade de inadimplência usando o status de estudante. `student` é codificado como uma variável dummy, com um valor de 1 para um estudante e um valor de 0 para um não-estudante e representado pela variável `studente[Yes]` na tabela.

e o p-valor associado é estatisticamente significativo. Isso indica que os estudantes tendem a ter maiores probabilidades de inadimplência do que os não estudantes:

$$\hat{Pr}(default = Yes|student = Yes) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431$$

$$\hat{Pr}(default = Yes|student = Yes) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292$$

Lista I - Regressão Logística

Regressão Logística no R

Uma população de mulheres, com pelo menos 21 anos de idade, da etnia indígena Pima que vivia perto de Phoenix, Arizona, foi testada para diabetes de acordo com os critérios da Organização Mundial de Saúde. Os dados foram coletados pelo Instituto Nacional de Diabetes e Doenças Digestivas e Renais dos EUA. Os 532 registros completos estão em dois data frames disponibilizados pelo pacote R `MASS`. O data frame `pima.te` é uma amostra de teste com 200 registros escolhidos aleatoriamente e o `pima.tr` é uma base de treinamento com os demais 332 registros. Os dados coletados nesse trabalho foram:

- `npreg` - número de gravidez.
- `glu` - concentração de glicose plasmática em um teste oral de tolerância à glicose.
- `pb` - pressão arterial diastólica (mm Hg).
- `skin` - espessura da dobra cutânea tricipital (mm).
- `bmi` - índice de massa corporal (peso em kg / (altura em m^2).
- `ped` - função de pedigree de diabetes.
- `age` - idade em anos.
- `type` - Yes ou No, para diabéticos, de acordo com os critérios da OMS.

Execute os comandos abaixo dos exercícios abaixo:

Exercício nº 1

Crie o data frame `teste` com os dados de `pima.te` e o data frame `treino` com a com os dados de `pima.tr`. Altere a codificação da nossa variável de interesse (`type`) em:

- 0 = Não-diabético, e
- 1 = Diabético.

```
library(MASS)
library(ggplot2)

treino <- Pima.tr
teste  <- Pima.te
treino$type <- ifelse(treino$type=="Yes",1,0)
teste$type <- ifelse(teste$type=="No",0,1)
# Missing values?
cat("# valores missing?",'\n')
sapply(treino, function(x) sum(is.na(x)))
```

Exercício nº 2

Faça gráficos de dispersão entre todas as variáveis explicativas com a função `pairs()`, e altere as cores dos pontos de acordo com a ocorrência ou não de diabetes (variável `type`). Além disso, tente fazer um gráfico da variável `type` em função da idade (coluna `age`). Use a função `jitter()` para tornar seu gráfico mais informativo. Adicione um ajuste logístico baseado na idade em cima do gráfico?

```
# Scatterplot matrix
pairs(subset(treino, select = -c(type)), col = as.factor(treino$type))

# Dispersão da Idade versus a Ocorrência de Diabete
  ggplot(treino, aes(x = age, y = type)) +
  geom_jitter(width = 0.5, height = 0.07, alpha = .2) +
  geom_smooth(method = "glm", se = FALSE,
              method.args = list(family = "binomial")) +
  labs(y = expression(hat(P)(Diabetica)))
```

Exercício nº 3

Use a função `glm()` e com os dados do data frame `treino`, faça um ajuste com base no modelo logístico da variável `type` em função da idade (coluna `age`) e coluna Índice de massa corporal (variável `bmi`). Imprima os coeficientes e seu `p-value` do modelo ajustado.

```
modelo1 <- glm(type ~ age + bmi, data = treino, family = binomial)
summary(modelo1)
```

Exercício nº 4

Qual a previsão do modelo ajustado no exercício 3, em termos de probabilidade, para alguém com a idade 35 e IMC de 32? E o que diz se IMC for 22?

```
# Usando a função predict()
predict(modelo1, type = "response", newdata = data.frame(bmi = c(32, 22), age = 35))

# Através da função logística
lgs_fun <- function(par, x)
{
  1 / (1 + exp(-x \%*\% par))
  # x \%*\% par é  equivalente a formula b\_0 + b\_1*age + b2*bmi
}

lgs_fun(modelo1$coefficients, c(1, 35, 32))
lgs_fun(modelo1$coefficients, c(1, 35, 22))
```

Exercício nº 5

De acordo com o modelo, quais são as chances (`odds`) de uma mulher em nossa amostra ser diabética, aos 55 anos e 37 anos? Lembre-se que quando falamos em chances (`odds`) neste contexto têm uma definição muito bem clara e que é diferente da probabilidade.

```
# Usando a definicão de Logit
lgs_fun(modelo1$coefficients, c(1, 55, 37)) /
       (1 - lgs_fun(modelo1$coefficients, c(1, 55, 37)))

# ou usando um forma mais simples, com a notações de algebra linear
exp(c(1, 55, 37) %*% modelo1$coefficients)

# Or através da função definida do R
exp(predict(modelo1, response = "link", newdata = data.frame(age = 55, bmi = 37)))
```

Exercício nº 6

Construa a matriz de confusão, entre os valores observados de `type` no data frame `treino` e a previsão do modelo. Use um valor de corte de 0,5, isto é que as mulheres cujo modelo estima ter pelo menos 50% chance são consideradas diabéticas. Qual é a precisão (**acuracia**) das previsões?

```
mc1 <- table(treino$type[!is.na(treino$bmi)],
             predict(modelo1, type = "response") >= 0.5)
mc1

# Calcula do a precisão (Acurácia das predições)
sum(diag(mc1)) / sum(mc1)
```

Exercício nº 7

Agora construa a matriz de confusão para o conjunto de dados de teste. Qual é a precisão (**acuracia**) das previsões?

```
# Verificar se o conjunto de teste esta com missing values
sapply(teste, function(x) sum(is.na(x)))

# Fazer as predições
teste_pred <- predict(modelo1, type = "response", newdata = teste)

# ou manualmente (através das definuções)
teste_predm <- lgs_fun(modelo1$coefficients, as.matrix(cbind(1, teste[, c("age", "bmi")])))

mc1_teste <- table(teste$type,teste_pred >= 0.5)
mc1_teste

# Calculando a precisão (Acurácia das predições)
sum(diag(mc1_teste)) / sum(mc1_teste)
```