Prova Estatística Bayesiana

André Santos

04 de dezembro de 2020

- Universidade: Universidade Nove de Julho
- Programa: Pós-graduação em Administração
- Professor: PhD José Eduardo Storopoli
- Aluno: André Luis Marques Ferreira dos Santos
- RA: ‘620150027’
- Disciplina: Estatística Bayesiana

Estudo de caso:

Os gestores de um e-commerce de moda feminina precisam entender quais os fatores que influenciam as vendas.A amostra é do ano de 2005 e contém informações de 730 clientes de diversas regiões do Brasil.As variáveis de interesse são gênero, frequência de compras no mês, se as pessoas moram na capital, formas de pagamento, quantidade de produtos e valor do pedido.

Objetivo geral:

Aplicar um modelo de regressão linear generalizado - Binomial em um relatório de vendas de um e-commerce de moda feminina para inferir sobre propensão à compra.

Objetivos específicos:

- Inferir sobre quais variáveis têm mais influência nas vendas
- Prever quais clientes têm maior propensão à compra
- Comparar o modelo bayesiano com um modelo matemático de otimização não linear

Justificativa na escolha do modelo:

Foi aplicado o modelo binomial sobre os dados pois o objetivo é determinar se os clientes irão ou não comprar no site. Ou seja, nosso problema é binário (compra ou não compra)

Amostra

- Tamanho: 1026 observações (50% treino; 50% teste)
- Total de clientes (**clientes**): número de identificação do cliente na amostra
- Status da compra (**status**): realizou a compra: 1; desistiu da compra: 0
- Gênero (**genero**): homem = 1; mulher =0
- Frequência de compras no mês (**compras**): total de pedidos que uma pessoa realizou em um único mês
- Região (**capital**): capital = 1; interior = 0
- Forma de pagamento (**pagto**): crédito = 1; boleto = 0
- Valor do pedido (**pedido**): valor do pedido com frete
- Quantidade (**qtde**): quantidade de produtos adquiridos por pedido

Dicionário de variáveis

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
Variavel <- c("clientes", "status", "genero", "compras", "capital", "pagto", "pedido", "qtde")
Valor <- c("ID cliente", "Comprou = 1; desistiu = 0", "Homem = 1, mulher = 0", "Total de compras no mês",
           "Mora na capital = 1, mora no interior = 0", "Pagou com crédito = 1; pagou no boleto = 0",
           "Valor do pedido","Quantidade de produtos no pedido")
tabela <- cbind(Variavel, Valor)
tabela %>%
  knitr::kable()
```

Variavel	Valor

Variavel	Valor
clientes	ID cliente
status	Comprou = 1; desistiu = 0
genero	Homem = 1, mulher = 0
compras	Total de compras no mês
capital	Mora na capital = 1, mora no interior = 0
pagto	Pagou com crédito = 1; pagou no boleto = 0
pedido	Valor do pedido
qtde	Quantidade de produtos no pedido

```
library (readr)
urlfile="https://raw.githubusercontent.com/andrelmfsantos/Bayes_MLG/main/dataset_prova_bayes.csv"
mydata<-read_csv2(url(urlfile))
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Parsed with column specification:
## cols(
##   clientes = col_double(),
##   status = col_double(),
##   genero = col_double(),
##   compras = col_double(),
##   capital = col_double(),
##   pagto = col_double(),
##   pedido = col_double(),
##   qtde = col_double()
## )
```

```
head(mydata)
```

```
## # A tibble: 6 x 8
##   clientes status genero compras capital pagto pedido  qtde
##      <dbl>  <dbl>  <dbl>   <dbl>   <dbl> <dbl>  <dbl> <dbl>
## 1        1      0      0       1       1     0    400     2
## 2        2      0      0       1       0     1   1000     1
## 3        4      0      1       1       0     0    200     1
## 4        5      0      0       1       0     0    500     5
## 5        6      1      0       2       0     0    400     2
## 6        8      1      0       2       0     0    400     2
```

Resumo estatístico

```
summary(mydata[-1])
```

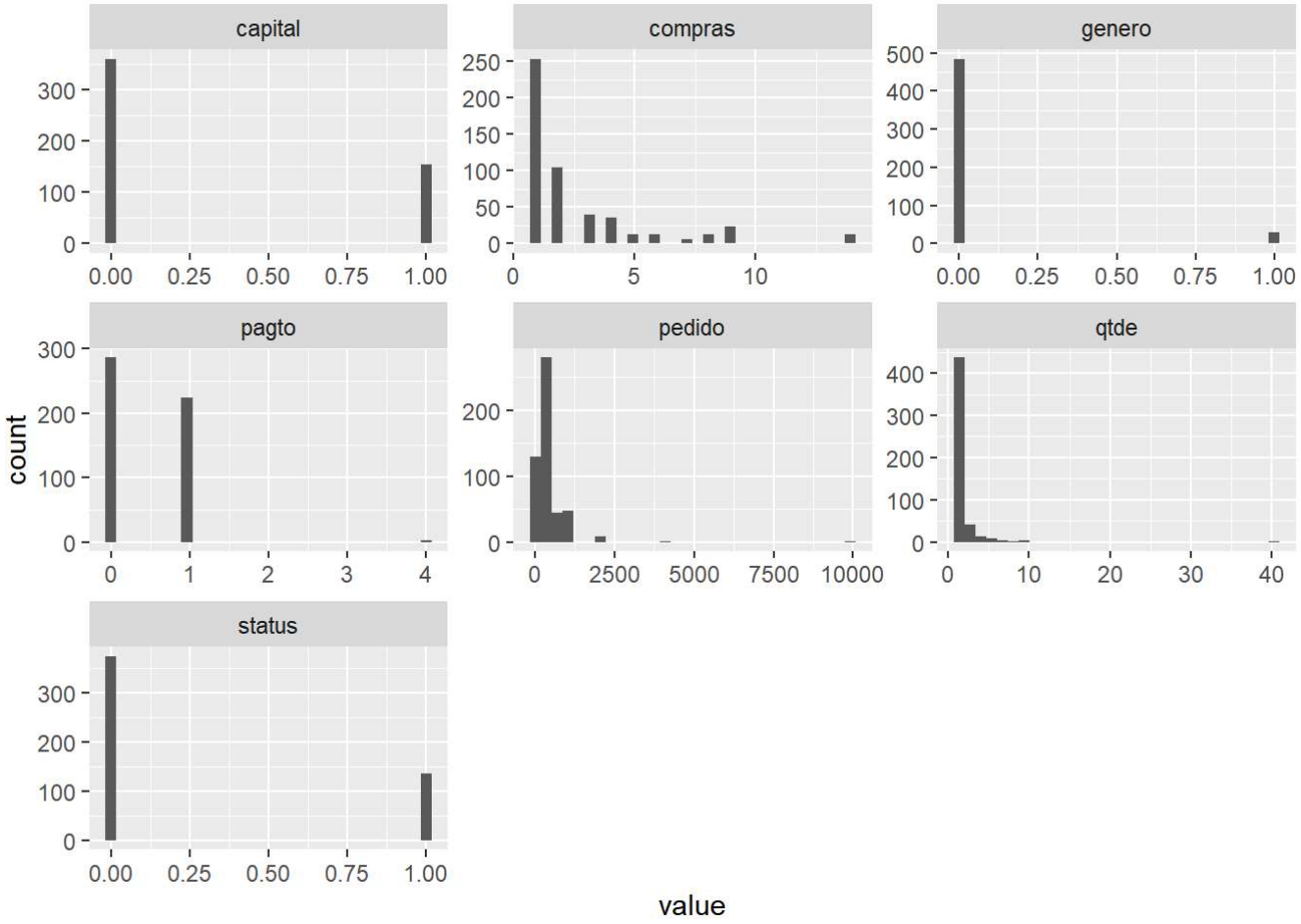
```
##      status           genero           compras          capital      
##  Min.   :0.0000   Min.   :0.00000   Min.   : 1.000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.: 1.000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.00000   Median : 2.000   Median :0.0000  
##  Mean   :0.2676   Mean   :0.05469   Mean   : 2.707   Mean   :0.2988  
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.: 3.000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :1.00000   Max.   :14.000   Max.   :1.0000  
##      pagto            pedido           qtde       
##  Min.   :0.0000   Min.   :   90   Min.   : 1.000  
##  1st Qu.:0.0000   1st Qu.:  100   1st Qu.: 1.000  
##  Median :0.0000   Median :  300   Median : 1.000  
##  Mean   :0.4531   Mean   :  409   Mean   : 1.711  
##  3rd Qu.:1.0000   3rd Qu.:  500   3rd Qu.: 2.000  
##  Max.   :4.0000   Max.   :10000   Max.   :40.000
```

Histogramas das variáveis

```
library(purrr)
library(tidyr)
library(ggplot2)
library(dplyr)

charts <- select(mydata, capital, compras, genero, pagto, pedido, qtde, status)
charts %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) + facet_wrap(~ key, scales = "free") + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Regressão logística com o rstanarm

Modelo

```
# Modelo
options(mc.cores = parallel::detectCores())
options(Ncpus = parallel::detectCores())

library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.21.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
model_binomial <- stan_glm(
  status ~ genero + compras + capital + pagto + pedido + qtde,
  data = mydata,
  family = binomial()
    )
```

Resumo do modelo

```
summary(model_binomial)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      status ~ genero + compras + capital + pagto + pedido + qtde
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 512
##  predictors:   7
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -1.5    0.2 -1.8  -1.5  -1.2
## genero       0.6    0.4  0.0   0.6   1.1
## compras      0.1    0.0  0.0   0.1   0.1
## capital     -0.4    0.2 -0.7  -0.4  -0.1
## pagto        0.6    0.2  0.3   0.6   0.8
## pedido       0.0    0.0  0.0   0.0   0.0
## qtde        -0.1    0.1 -0.2  -0.1   0.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.3    0.0  0.2   0.3   0.3
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for details see help('summa
ry.stanreg')).
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  4571
## genero        0.0  1.0  5011
## compras       0.0  1.0  5115
## capital       0.0  1.0  5430
## pagto         0.0  1.0  5251
## pedido        0.0  1.0  2910
## qtde          0.0  1.0  2971
## mean_PPD      0.0  1.0  4447
## log-posterior 0.0  1.0  1855
```

```
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, and Rhat is th
e potential scale reduction factor on split chains (at convergence Rhat=1).
```

Resultados

1.Coeficientes

```
coeff <- exp(model_binomial$coefficients)
coeff
```

```
## (Intercept)      genero     compras     capital       pagto      pedido
##   0.2248296   1.8072924   1.0661963   0.6928158   1.8097072   1.0005059
##        qtde
##   0.9232469
```

2.Interpretação dos coeficientes

```
library(dplyr)
Coeficientes <- c("Intercepto", "genero (beta 1)", "compras (beta 2)", "capital (beta 3)", "pagto (beta 4)", "pedido (beta
 5)",
                  "qtde (beta 6)")
Analise <- c("Dadas todas outras variáveis com valores nulos temos a chance de uma pessoa comprar na loja = 22,1%",
             "Dado que um cliente é homem aumenta a chance dele comprar na loja em 81%",
             "Conforme aumenta a frequência de pedidos aumenta a chance de comprar em 6,5%",
             "Se a pessoa mora na capital cai a chance de comprar em 30,1%",
             "Pedidos feitos com cartão de crédito aumenta as chances de realizar a compra em 81,9%",
             "Conforme aumenta o valor do pedido aumenta a chance de realizar uma compra em 0,05%",
             "Cada produto a mais adicionado ao carrinho cai a chance de compra em 7,6%")
# coeficientes:
#(Intercept)      genero     compras     capital       pagto      pedido        qtde
#  0.2219928   1.8106876   1.0658785   0.6993211   1.8196994   1.0005085   0.9248157

result <- cbind(Coeficientes, Analise)
result %>%
  knitr::kable()
```

Coeficientes	Analise
Intercepto	Dadas todas outras variáveis com valores nulos temos a chance de uma pessoa comprar na loja = 22,1%
genero (beta 1)	Dado que um cliente é homem aumenta a chance dele comprar na loja em 81%
compras (beta 2)	Conforme aumenta a frequência de pedidos aumenta a chance de comprar em 6,5%
capital (beta 3)	Se a pessoa mora na capital cai a chance de comprar em 30,1%
pagto (beta 4)	Pedidos feitos com cartão de crédito aumenta as chances de realizar a compra em 81,9%
pedido (beta 5)	Conforme aumenta o valor do pedido aumenta a chance de realizar uma compra em 0,05%
qtde (beta 6)	Cada produto a mais adicionado ao carrinho cai a chance de compra em 7,6%

Considerações

O gênero e a forma de pagamento são as variáveis que mais impactam na conversão em compras. É esperado que peidos com valores altos tenham maiores chances de serem efetivos em compras, do que pedidos com valores mais baixos (média de pedidos R$300). Residir na capital ou conforme as pessoas adicionam mais produtos no carrinho diminuem as chances de efetivação dos pedidos em compras. Este estudo também teve o objetivo de comparar a abordagem bayesiana com a frequentista. Tanto na base de treino, quanto na base de teste o modelo bayesiano foi superior, apresentando menor erro e melhor poder de predição, conforme pode ser observado na tabela a seguir.

```
Bayesiano <- c("192.551", "177.432")
Frequentista <- c("202.677", "194.213")
result <- cbind(Bayesiano, Frequentista)
rownames(result) <- c("Erro no Treino", "Erro no Teste")
result %>%
  knitr::kable()
```

	Bayesiano	Frequentista
Erro no Treino	192.551	202.677
Erro no Teste	177.432	194.213

Referências

- storopoli.github (https://storopoli.github.io/Estatistica-Bayesiana/6-Regressao_Binomial.html)