Guilherme Souza

Regressão logística multinomial em dados de e-commerce brasileiros

Niterói

2018

Guilherme Souza

Regressão logística multinomial em dados de e-commerce brasileiros

Trabalho apresentado como requisito para conclusão do curso de Econometria.

Universidade Federal Fluminense – UFF

Faculdade de Administração e Ciências Contábeis – EST

Departamento de Administração – STA

Programa de pós-graduação em Administração

Orientador: Prof. Dr. Eduardo Camilo da Silva

Niterói

2018

Regressão logística multinomial em dados de e-commerce brasileiros

14 de outubro de 2018

Os Provedores

Os dados a serem utilizados neste relatório foram publicados pela Olist[1], uma loja que atua dentro dentro dos marketplaces. É uma grande loja de departamentos dos principais e-commerces do Brasil. Conecta micro, pequenas e médias empresas (PMEs) a esses principais marketplaces por meio de contratos.

Os dados são provenientes de transações comerciais reais[2] envovelndo clientes e vendedores, sendo que a referência aos nomes das empresas vendedoras (parceiros da Olist) foram trocados por nomes de casas de Game of Thrones.

Quando um cliente compra um produto da Olist, o vendedor é notificado para atender ao pedido. Assim que o cliente recebe o produto, ou a data prevista da entrega vence, o cliente recebe uma pesquisa de satisfação por email onde ele pode dar uma nota pela experiência de compra e escrever alguns comentários.

Os datasets estão publicados na plataforma Kaggle[3], uma grande comunidade voltada à análise de dados e aprendizado de máquina.

O Dataset

Originalmente, a fonte de dados inclui 6 bases `.csv` distintas, cada qual destinadas a estudos específicos, mas todos são relacionados a eventos de compra em ecommerce pela plataforma Olist . A base utilizada neste relatório corresponde ao arquivo `.csv` de nome `olist_classified_public_dataset_v2`. Nesta base de dados, cada registro corresponde a uma compra de um cliente, referente a um produto e uma análise/crítica. Na tabela 1, vemos um `head()` das 6 primeiras colunas do dataset utilizando o ambiente `kableExtra`[4]:

Variáveis

O dataset é composto por 3,584 linhas e 34 colunas das quais 9 são destinas à composição da coluna `most_voted_class`. O valor dos campos desta coluna refletem o resultado agregado de dados originários de uma pesquisa de satisfação enviada ao cliente após a concretização da compra. Ao responder à pesquisa, o cliente possibilita, por intermádio dos dados, a criação de variáveis "intermediárias", também presentes na base de estudo e identidicadas pelo prefixo `votes_*`. O valor presente nas mesmas é inteiro e corresponde a uma pontuação que varia de 0 a 6. Estas variávies representam a intensidade do evento identificado no nome da coluna correspondente. A definição de cada uma delas é assim dada:

- **votes_before_estimate**: votos recebidos para entrega antes das mensagens de data estimadas.
- **votos_delayed**: votos recebidos por reclamações atrasadas.
- **votes_low_quality**: votos recebidos por reclamações de baixa qualidade do produto.
- **votes_return**: votos recebidos por desejar devolver o produto às reclamações do vendedor.
- **votes_not_as_anounced**: votos recebidos por produto não como reclamações anunciadas.
- **votes_partial_delivery**: votos recebidos por reclamações de entrega parcial (nem todos os produtos entregues).

[1] https://olist.com/

[2] https://www.kaggle.com/olistbr/brazilian-ecommerce#olist_public_dataset_v2.csv

[3] https://www.kaggle.com/

[4] `kableExtra` auxilia na construção de tabelas complexas e permite encadear o código com o comando `kable` e sintaxa do tipo pipe. Escrito por ZHU (2018), encontra-se na versão 0.9.0.

Table 1: Inspecionando o dataset

X	id	order_status	order_products_value	order_freight_value	order_items_qty
0	1	delivered	89.99	14.38	1
1	2	delivered	69.00	15.23	1
2	3	delivered	99.80	15.86	2
3	4	delivered	87.00	12.74	1
4	5	delivered	99.90	17.95	1
5	6	delivered	39.99	0.15	1

- **votes_other_delivery**: votos recebidos para outras reclamações relacionadas à entrega.
- **votes_other_order**: votos recebidos por outras reclamações relacionadas ao pedido.
- **votes_satisfied**: votos recebidos para mensagens satisfeitas pelo cliente.

Baseado no resultado agregado destas variáveis, a primeira variável intermediária, `most_voted_sub_class` é criada. É uma variável categórica cujos níveis são:

- **antes_prazo**: produto chegou antes do prazo estimado.
- **atrasado**: produto chegou depois do prazo estimado.
- **baixa_qualidade**: produto avaliado como de baixa qualidade.
- **devolucao**: comprador tem a intenção de devolver o produto.
- **diferente_do_anunciado**: em desacordo com o produto anunciado.
- **entrega_parcial**: veio com algum componente faltando.
- **outro_entrega**: será feita uma nova entrega.
- **outro_pedido**: será feito um novo pedido.
- **satisfeito**: comprador satisfeito com o produto.

Por fim, a partir do resultado desta variável é possível gerar a classificação resultante para o campo `most_voted_class`. Em outras palavras, temos a variável resposta . Os valores possíveis são:

- **problemas_de_entrega**
- **problemas_de_qualidade**
- **satisfeito_com_pedido**

As demais variáveis são:

- **index**: coluna de índice.
- **id**: variável inteira que faz referência a linha.
- **order_status**: referência ao status da ordem (delivered, shipped, etc)

Table 2: Status das ordens

order_status	count	percent
delivered	3467	0.9673549
shipped	48	0.0133929
canceled	25	0.0069754
invoiced	24	0.0066964
processing	20	0.0055804

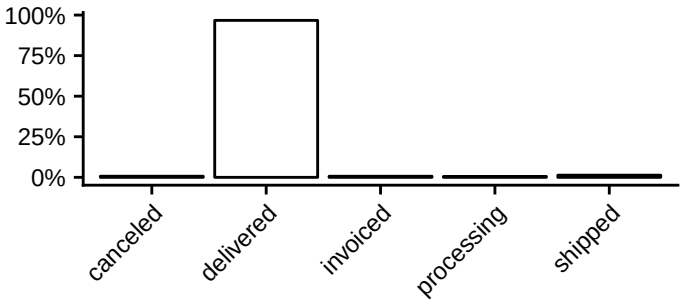


Figure 1: Status das ordens

- **order_products_value**: preço total de uma ordem de compra.
- **order_freight_value**: valor total do frete em um pedido.
- **order_sellers_qty**: quantidade total de vendedores que atenderam a um pedido.

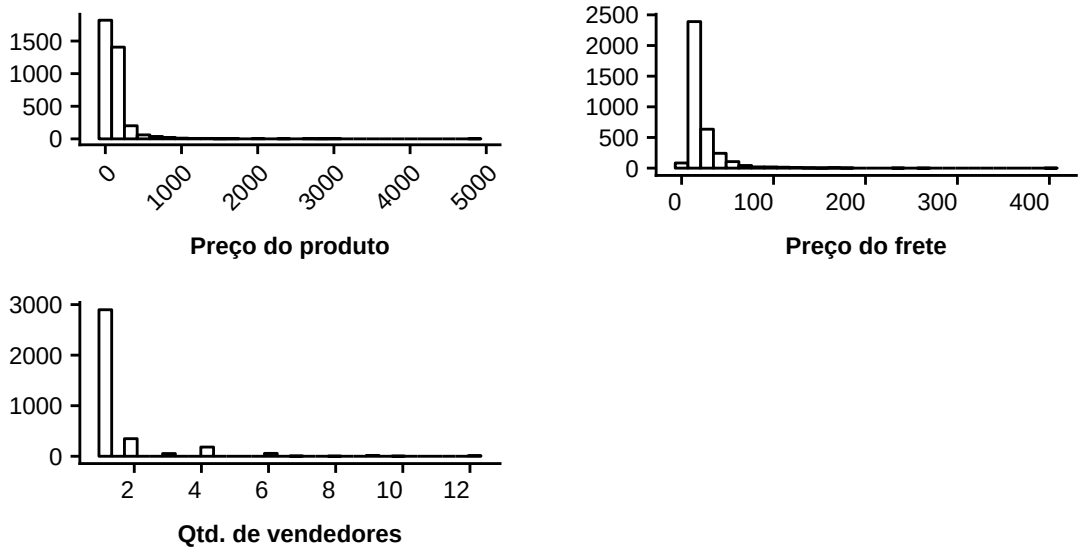


Figure 2: Preço, frete e quantidade de vendedores que atenderam uma ordem.

- **order_items_qty**: quantidade total de itens comprados em um pedido.

Table 3: Total de itens

order_items_qty	count	percent
1	3337	93.1%
2	180	5.0%
3	39	1.1%
4	15	0.4%
5	6	0.2%
6	6	0.2%
8	1	0.0%

- **order_purchase_timestamp**: mostra o registro de data e hora da compra.

Table 4: Qtd. de ordens em 2017 e 2018

year	count	percent
2017	2556	71.3%
2018	1028	28.7%

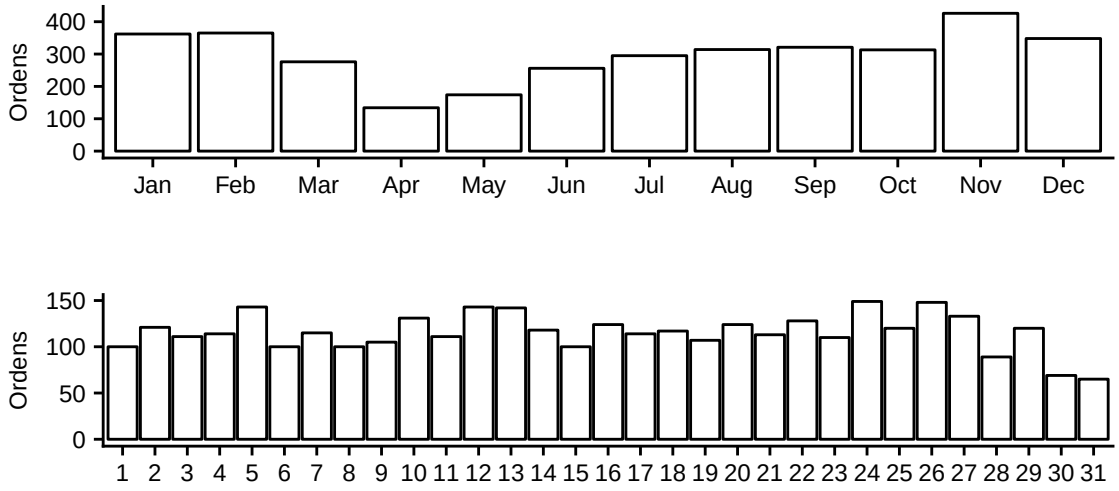


Figure 3: Estatísticas mensais e diárias das ordens de compras.

- **order_aproved_at**: mostra o carimbo de data / hora da aprovação do pagamento.

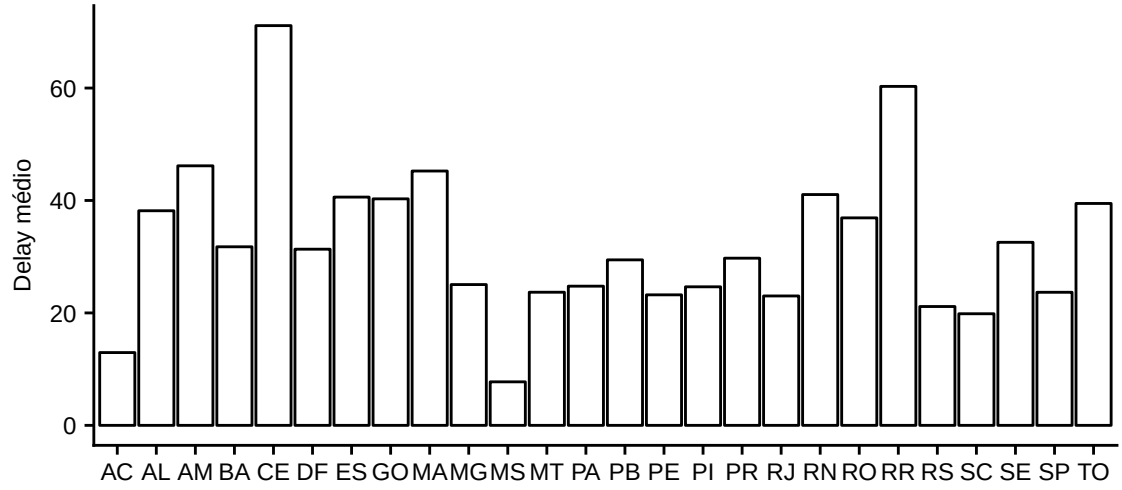


Figure 4: Delay médio por UF calculado como a diferença entre as datas de aprovação e ordem de compra. Mato grosso e Acre são os destaques com o menor tempo de aprovação de ordens.

- **order_estimated_delivery_date**: mostra a data de entrega estimada que foi informada ao - cliente no momento da compra.
- **order_delivered_customer_date**: mostra a data real de entrega do pedido ao cliente.

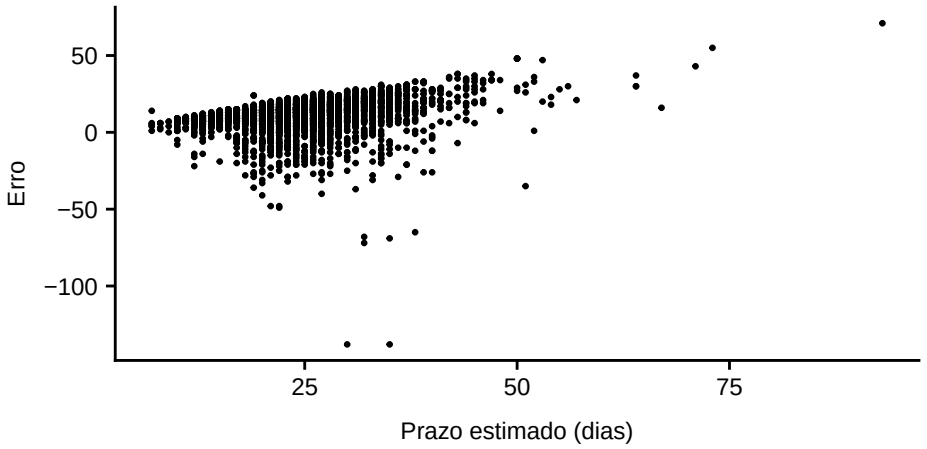


Figure 5: Comparação entre a estimação do prazo e o erro.

- **customer_city**: cidade do cliente
- **customer_zip_code_prefix**: os três primeiros dígitos do código postal do cliente. A figura 6 exibe uma tentativa de construção de visualização espacial de dados com auxilio do pacote `leaflet`. O pacote permite a criação de mapas interativos que podem ser utilizados diretamente do console do R (CHENG; KARAMBELKAR; XIE, 2018). Cada ponto no mapa representa uma cidade e uma ordem. A localização de cada ponto se deu por sucessivas chamadas à API de geolocalização *Data Science Toolkit*[5] pela função `geocode()` do pacote `ggmap`. Este pacote oferece um extenso conjunto de ferramentas que permite acessar conteúdo estático de provedores como Google Maps, OpenStreetMap, entre outros. (KAHLE; WICKHAM, 2013)

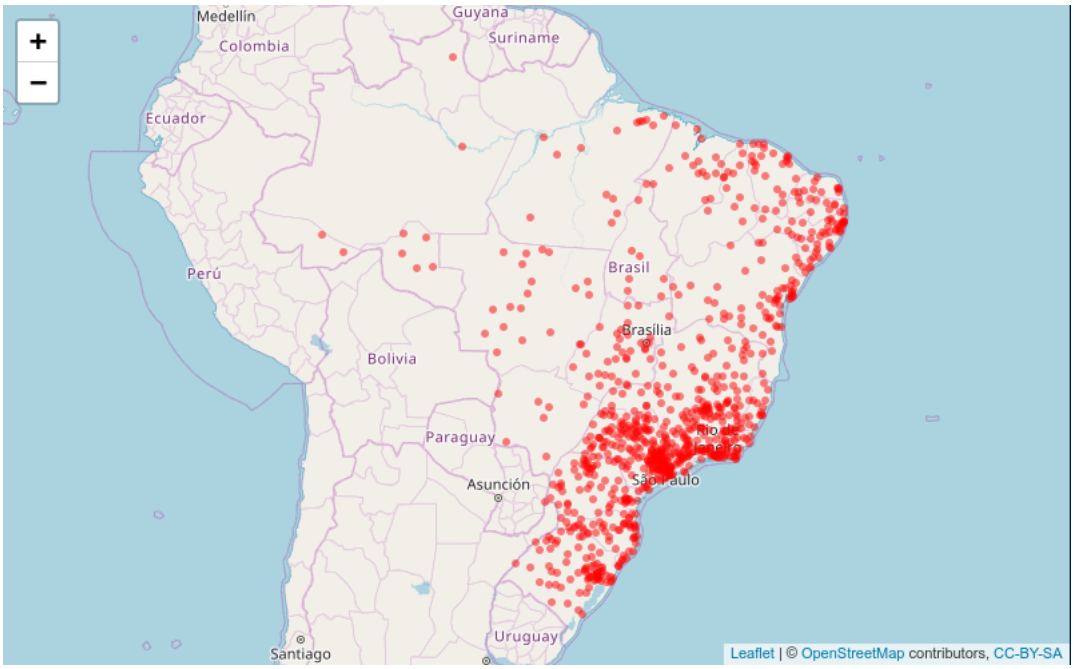


Figure 6: Visualização espacial da frequência de ordens por cidades brasileiras com utilização do pacote leaflet

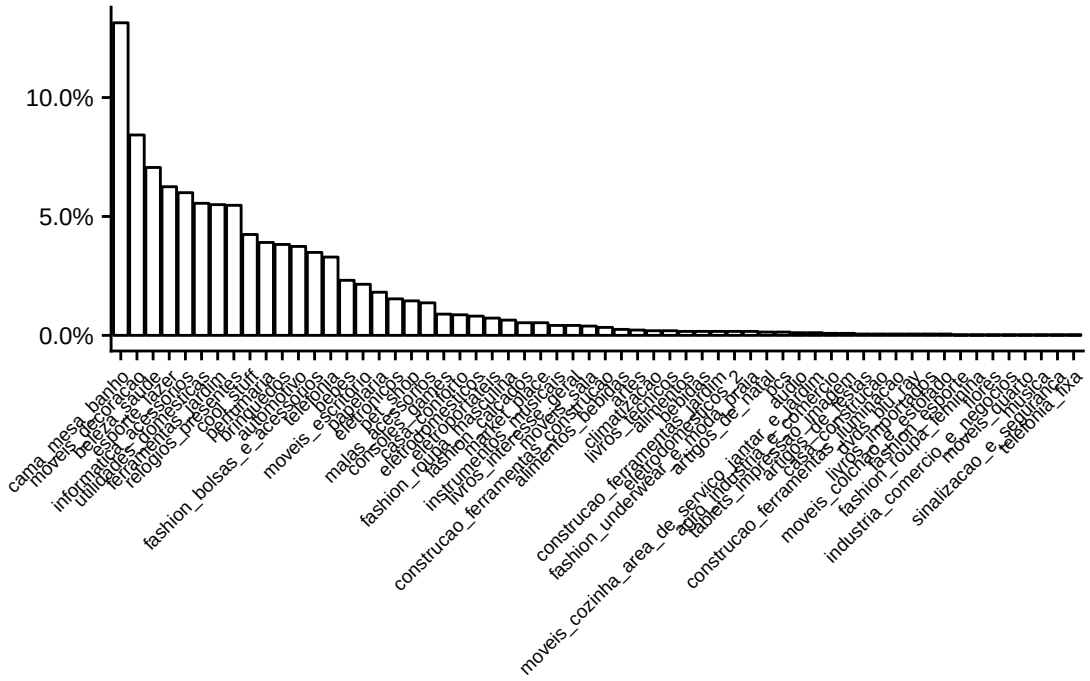
- **customer_state**: estado / província do cliente

[5] http://www.datasciencetoolkit.org/

Table 5: Frequência de ordens por estado

customer_state	count	percent
SP	1369	38.2%
RJ	505	14.1%
MG	438	12.2%
RS	188	5.2%
PR	161	4.5%
BA	150	4.2%
SC	118	3.3%
GO	84	2.3%
PE	76	2.1%
DF	74	2.1%
ES	70	2.0%
CE	58	1.6%
PA	58	1.6%
MT	42	1.2%
MA	41	1.1%
AL	24	0.7%
PI	24	0.7%
PB	23	0.6%
RN	21	0.6%
MS	18	0.5%
SE	16	0.4%
TO	10	0.3%
RO	7	0.2%
AM	5	0.1%
AC	3	0.1%
RR	1	0.0%

- **product_category_name**: a categoria raiz do produto adquirido, em português.



- **product_name_lenght**: número de caracteres extraídos do nome do produto comprado.
- **product_description_lenght**: número de caracteres extraídos da descrição do produto - adquirido.
- **product_photos_qty**: número de fotos publicadas de produtos comprados.
- **review_score**: nota variando de 1 a 5 dada pelo cliente em uma pesquisa de satisfação.

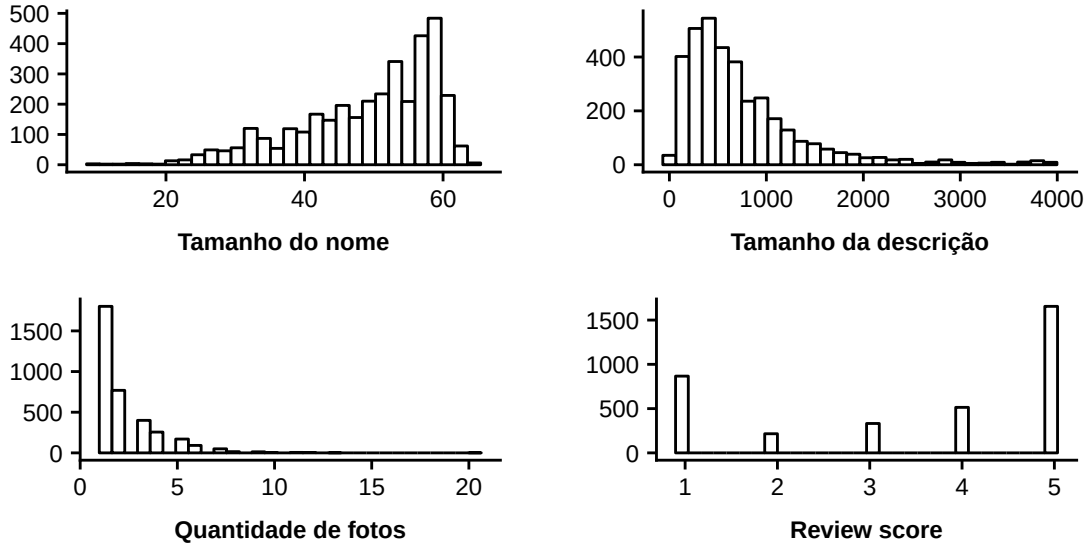


Figure 7: Estatísticas do anúncio na seguinte ordem, tamanho do nome em caracteres, tamanho da descrição em caracteres, quantidade de fotos e avaliação obtida.

- **review_creation_date**: mostra a data em que a pesquisa de satisfação foi enviada ao cliente.
- **review_answer_timestamp**: mostra o registro de data e hora da resposta da pesquisa de satisfação.

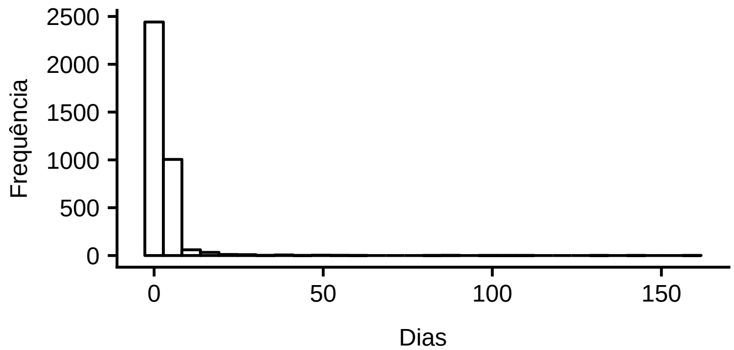


Figure 8: Demora para responder a pesquisa de safistação. É a diferença entre a data de envio da pesquisa de satisfação e a data de resposta.

- **review_comment_title**: título do comentário da resenha deixada pelo cliente, em - português.
- **review_comment_message**: mensagem de comentário da avaliação deixada pelo cliente, em português. A figura 9 mostra a nuvem de palavras criada com auxilio do pacote `tm` que fornece insfraestrutura para mineração de textos (FEINERER; HORNIK; MEYER, 2008).

Figure 9: Word cloud de palavras mais comuns presentes nos comentários sobre os produtos. Extraído da coluna review.comment.message.

Modelos e Regressões

Regressão Logística Multinomial

Para a criação de um modelo de regressão múltipla, a variável mais interessante seria `most_voted_class`. Neste caso, poderíamos ter uma equação para estimar em que classe se encontra uma ordem de compra com base em um conjunto de variáveis preditoras. Contudo, trata-se de uma variável categórica com 3 níveis.

Uma alternativa é fazer uma regressão logística multinomial (RLM). Trata-se de uma forma de regressão condizuda quando a variável dependente é categórica com mais de dois níveis. É usada para descrever dados e explicar a relação entre uma variável nominal dependente e uma ou mais variáveis independentes contínuas. As variáveis nominais podem ser entendidas como variáveis que não possuem uma ordem intrínsceca. De fato, se tomássemos como variável resposta `most_voted_class`, não poderíamos afirmar que existe uma ordem entre os nívies `problemas_de_entrega`, `problemas_de_qualidade`, e `satisfeito_com_pedido`. Desta forma, em teoria, podemos utilizar essa técnica para determinar, dado um conjunto de preditoras – grupo de variáveis relacionadas ao questionário, por exemplo, em qual classe se encontrará um cliente. Poderíamos utilizar tal modelo para determinar, por exemplo, a probabilidade de um cliente estar `satisfeito_com_pedido` utilizando os dados relativos a sua compra.

Para a realização da análise de regressão nos dados de e-commerce da Olist seguiu-se os procedimentos do tutorial disponível em Analytics Vidhya (2016).

Após as modelagens, algumas variáveis auxiliares foram criadas, como `order_purchase_aprove_delay`, `delivery_delay`, `delivery_estimation_delay`, `order_delivery_estimation_delay_error`. Abaixo, a função `str()` apresenta a estrutura do dataset resultante onde será aplicada a RLM:

```
## 'data.frame':    3584 obs. of  25 variables:
##  $ order_products_value                : num  90 69 99.8 87 99.9 ...
##  $ order_freight_value                 : num  14.4 15.2 15.9 12.7 17.9 ...
##  $ order_items_qty                     : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ order_sellers_qty                   : int  1 1 4 1 2 6 1 1 1 1 ...
##  $ customer_zip_code_prefix            : int  308 377 122 140 205 20 564 83 69 403 ...
##  $ product_name_lenght                 : int  59 50 59 45 60 53 54 50 47 60 ...
##  $ product_description_lenght          : int  492 679 341 411 189 386 1120 448 482 189 ...
##  $ product_photos_qty                  : int  3 4 2 1 1 1 8 1 2 1 ...
##  $ review_score                        : int  5 5 1 4 3 5 2 5 3 4 ...
##  $ votes_before_estimate               : int  0 3 0 0 0 0 0 2 1 0 ...
##  $ votes_delayed                       : int  0 0 0 3 0 0 0 0 0 0 ...
##  $ votes_low_quality                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ votes_return                        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ votes_not_as_anounced               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ votes_partial_delivery              : int  0 0 3 0 3 1 0 0 0 0 ...
##  $ votes_other_delivery                : int  0 0 0 0 0 1 3 0 0 0 ...
##  $ votes_other_order                   : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ votes_satisfied                     : int  3 0 0 0 0 0 0 1 2 3 ...
##  $ order_purchase_aprove_delay         : num  0.589 0.608 99.135 105.631 36.589 ...
##  $ delivery_delay                      : num  9 3 6 14 11 5 22 18 6 17 ...
##  $ delivery_estimation_delay           : num  22 28 19 35 26 19 31 19 20 28 ...
##  $ order_delivery_estimation_delay_error: num  13 25 13 21 15 14 9 1 14 11 ...
##  $ review_answer_delay                 : num  1 3 9 1 3 3 1 0 3 2 ...
##  $ most_voted_subclass                 : Factor w/ 10 levels "","antes_prazo",..: 10 2 7 3 7 1 8 2 
##  $ most_voted_class                    : Factor w/ 4 levels "","problemas_de_entrega",..: 4 4 2 2 2
```

No *chunk* abaixo vemos a modelagem do dataset utilizando sintaxe pipe do pacote `dplyr` (WICKHAM et al., 2018) e aplicação da RLM com utilização do pacote `nnet` (VENABLES; RIPLEY, 2002):

```
library(nnet)
olist_data_mod_reg <- olist_data_mod_reg %>%
    filter(most_voted_class != '')

olist_data_mod_reg$most_voted_class2 <- relevel(olist_data_mod_reg$most_voted_class,
                                                ref = "satisfeito_com_pedido")
rlm <- multinom(most_voted_class ~ ., data = olist_data_mod_reg[,c(9,18,20,          25)])
```

```
## Warning in multinom(most_voted_class ~ ., data = olist_data_mod_reg[,
## c(9, : group '' is empty
```

```
## # weights:  15 (8 variable)
## initial  value 3628.716389
## iter  10 value 1220.070480
## iter  20 value 1196.792440
## final  value 1196.732784
## converged
```

```
summary(rlm)
```

```
## Call:
## multinom(formula = most_voted_class ~ ., data = olist_data_mod_reg[,
##     c(9, 18, 20, 25)])
##
## Coefficients:
##                         (Intercept) review_score votes_satisfied
## problemas_de_qualidade  0.03302506   0.05167207       0.5850768
## satisfeito_com_pedido  -7.56304681   2.06762389       2.5684232
##                         delivery_delay
## problemas_de_qualidade    -0.05187714
## satisfeito_com_pedido     -0.09557719
##
## Std. Errors:
##                         (Intercept) review_score votes_satisfied
## problemas_de_qualidade   0.1452070   0.04614153       0.2081487
## satisfeito_com_pedido    0.5325287   0.11945811       0.2032135
##                         delivery_delay
## problemas_de_qualidade    0.006067884
## satisfeito_com_pedido     0.013087012
##
## Residual Deviance: 2393.466
## AIC: 2409.466
```

A RLM funciona como uma série de regressões logísticas, cada qual comparando dois níveis da variável resposta (`most_voted_class`).

1 - A função `relevel()` renivela os fatores da coluna da variável resposta fazendo com que `satisfeito_com_pedido` seja o fator de referência para as comparações feitas pela função.

2 - A aplicação da RLM neste dataset se dá por meio da função `multinom()` contida no pacote `nnet`, definindo como variável dependente `most_voted_class` e `review_score`, `votes_satisfied` e `delivery_delay` (colunas com índices 9, 18 e 20 respectivamente) como variáveis independentes. A saída da função fornece primeiramente um bloco de informações relacionadas a execução.

3 - São dois os blocos subsequentes na saída do `summary()`, o primeiro com os coeficientes e o próximo com as

estatísticas de erros padrão. No bloco *coefficients*, vemos que a primeira linha compara `problemas_de_entrega` com o fator de referência `satisfeito_com_pedido`. A segunda linha compara `problemas_de_qualidade` com o mesmo fator de referência e assim por diante.

Podemos também realizar testes com dados fictícios,

```
dummy_data <- data.frame(review_score = sample(c(1:5),5),
                         votes_satisfied = sample(c(1:6),5),
                         delivery_delay = sample(c(1:20),5))

pred <- predict(rlm, dummy_data, 'probs')

pred <- as.data.frame(pred) %>%
    cbind(dummy_data)

pred <- pred[,c(4,5,6,1,2,3)]

pred
```

```
##   review_score votes_satisfied delivery_delay problemas_de_entrega
## 1            3               5              9         2.437106e-05
## 2            4               3             18         1.234435e-03
## 3            2               2             10         1.878854e-01
## 4            5               1              2         5.681775e-03
## 5            1               4             19         4.071599e-02
##   problemas_de_qualidade satisfeito_com_pedido
## 1           0.0003437544             0.9996319
## 2           0.0035671557             0.9951984
## 3           0.4130562182             0.3990584
## 4           0.0123045818             0.9820136
## 5           0.1717362078             0.7875478
```

onde as três primeiras colunas são dados gerados de forma procedural e alimentados no modelo por meio da função `predict()`. Cada linha na tabela representa um cliente ou ordem. Podemos observar a variação da distribuição das probabilidades em relação a classe em que se encontraria o cliente conforme variam suas ações na pesquisa de satisfação e a demora na entrega do produto de modo que,

$$P_{prob.entrega} + P_{prob.qualidade} + P_{satisfeito} = 1$$

Estatísticas de validação

Abaixo vemos a saída da tabela de coeficientes e os intervalos de confiança de cada variável preditora:

```
##                          (Intercept) review_score votes_satisfied
## problemas_de_qualidade 1.0335764421     1.053030        1.795129
## satisfeito_com_pedido  0.0005192906     7.906015       13.045238
##                         delivery_delay
## problemas_de_qualidade       0.9494455
## satisfeito_com_pedido        0.9088482
```

```
## , , problemas_de_qualidade
##
##                     2.5 %    97.5 %
## (Intercept)     0.7775747 1.3738618
## review_score    0.9619780 1.1527009
## votes_satisfied 1.1937657 2.6994305
## delivery_delay  0.9382208 0.9608045
```

```
##
## , , satisfeito_com_pedido
##
##                         2.5 %       97.5 %
## (Intercept)     0.0001828611  0.001474687
## review_score    6.2556827661  9.991727353
## votes_satisfied 8.7594411091 19.427979196
## delivery_delay  0.8858326200  0.932461777
```

No primeiro bloco, cada linha apresenta os coeficientes de regressão multinomial para uma dada classe. O segundo e terceiro blocos contém as estatísticas sobre significância dos coeficiêntes das variáveis preditoras em relação a cada uma das possíveis classes de `most_voted_class`. Os intervalos de confiança assinalados vão de 2.5 % a 97.5 %.

A daterminação da significância da variável preditora na composição do modelo é feita com base na observação do intervalo de confiança. A variável terá significancia apenas se no intervalo de confiança que esteja situada, não apresentar 1 (EDUCATION, 2014). Neste caso, para `problemas_de_qualidade`, as variáveis significantes são `votes_satisfied` e `delivery_delay`. Para a segunda, incluímos todas as variáveis uma vez que todas são significantes.

Podemos agora comparar as variáveis significantes com os modelos do primeiro bloco e indicar as seguintes generalizações:

- Cada dia à mais de `delivery_delay` impacta inversamente a probabilidade de `problemas_de_qualidade` em comparação com `problemas_de_entrega`.
- Uma unidade de aumento em `votes_satisfied` impacta positivamente a probabilidade de o cliente estar `satisfeito_com_pedido` em comparação com ter tido `problemas_de_entrega`.
- Cada dia à mais de `delivery_delay` impacta inversamente a probabilidade de o cliente estar `satisfeito_com_pedido` em comparação com ter tido problemas na entrega.
- Cada unidade de aumento `satisfeito_com_pedido` aumenta as chances de o cliente estar `satisfeito_com_pedido` em comparação com ter tido `problemas na entrega`.

Referências

CHENG, J.; KARAMBELKAR, B.; XIE, Y. **leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library**. Traducao. [s.l: s.n.].

EDUCATION, Q. **(10) 4 Multinomial Logistic Regression - YouTube**, abr. 2014. Disponível em: <https://www.youtube.com/watch?v=zDIa2a4gTcE&t=656s>

FEINERER, I.; HORNIK, K.; MEYER, D. Text Mining Infrastructure in R. **Journal of Statistical Software**, v. 25, n. 5, p. 1–54, March 2008.

KAHLE, D.; WICKHAM, H. ggmap: Spatial Visualization with ggplot2. **The R Journal**, v. 5, n. 1, p. 144–161, 2013.

VENABLES, W. N.; RIPLEY, B. D. **Modern Applied Statistics with S**. Traducao. Fourth ed. New York: Springer, 2002.

WICKHAM, H. et al. **dplyr: A Grammar of Data Manipulation**. Traducao. [s.l: s.n.].

ZHU, H. **kableExtra: Construct Complex Table with 'kable' and Pipe Syntax**. Traducao. [s.l: s.n.].

Analytics Vidhya, fev. 2016. Disponível em: <https://www.analyticsvidhya.com/blog/2016/02/multinomial-ordinal-logistic-regression/>