Checkpoint3

Rodolfo Viana

04-06-2015

Utilizando os dados disponíveis no RecySys challenge 2015 (http://2015.recsyschallenge.com/challenge.html) tentamos classificar uma sessão como compradora ou não.

Os dados disponíveis podem ser encontrados aqui (http://2015.recsyschallenge.com/). No dataset inicial temos os seguintes atributos:

1. Session ID – o ID de uma sessão. Uma sessão possui um ou mais clicks
2. Timestamp – o tempo em que o click aconteceu
3. Item ID – o ID de um item
4. Category – a categoria do item
5. Buy – informação se o item foi comprado ou não

Como esses atributos eram atributos básicos começamos a criar novos atributos derivados dos atributos anteriores antes de montar um modelo de classificação. Atualmente temos os seguintes atributos:

1. SESSION - id da sessão
2. DAY - dia do acesso
3. MONTH - mês do acesso
4. TIME - hora do acesso, com minutos representados por quartis: {q1, q2, q3, q4}, exemplo a hora 14:48 é representada como 14.q4
5. ITEM - id do item
6. CATEGORY - id da categoria
7. WEEKDAY - dia da semana do acesso
8. CLICKED - quantidade de vezes que o item foi clicado (somando todos os usuários)
9. BOUGHT - quantidade de vezes que o item foi comprado (somando todos os usuários)
10. SOLDABILITY - razão de CLICKED por BOUGHT, multiplicado por 100
11. SAME_CAT - quantidade de produtos da categoria do click que também foi clicado pela sessão
12. SOLD_MEAN - média de vendabilidade dos item clicados pela sessão
13. SOLD_MEAN_DIFF - diferença, SOLDABILITY (do item) menos SOLD_MEAN (da sessão)
14. SOLD_MEDIAN - mediana das vendabilidades dos itens clicados pela sessão
15. SESSION_SIZE - número de clicks que a sessão deu
16. IS_BUY - 0 para para não compra e 1 para compra
17. CATEG_MOST - categoria de maior ocorrência do item
18. SESSION_DURATION - duração, em segundos, da sessão
19. RELATIVE_TIME = diferença de tempo entre o click e o primeiro click da sessão
20. RELATIVE_TIME_PROP = razão relative_time / session_duration

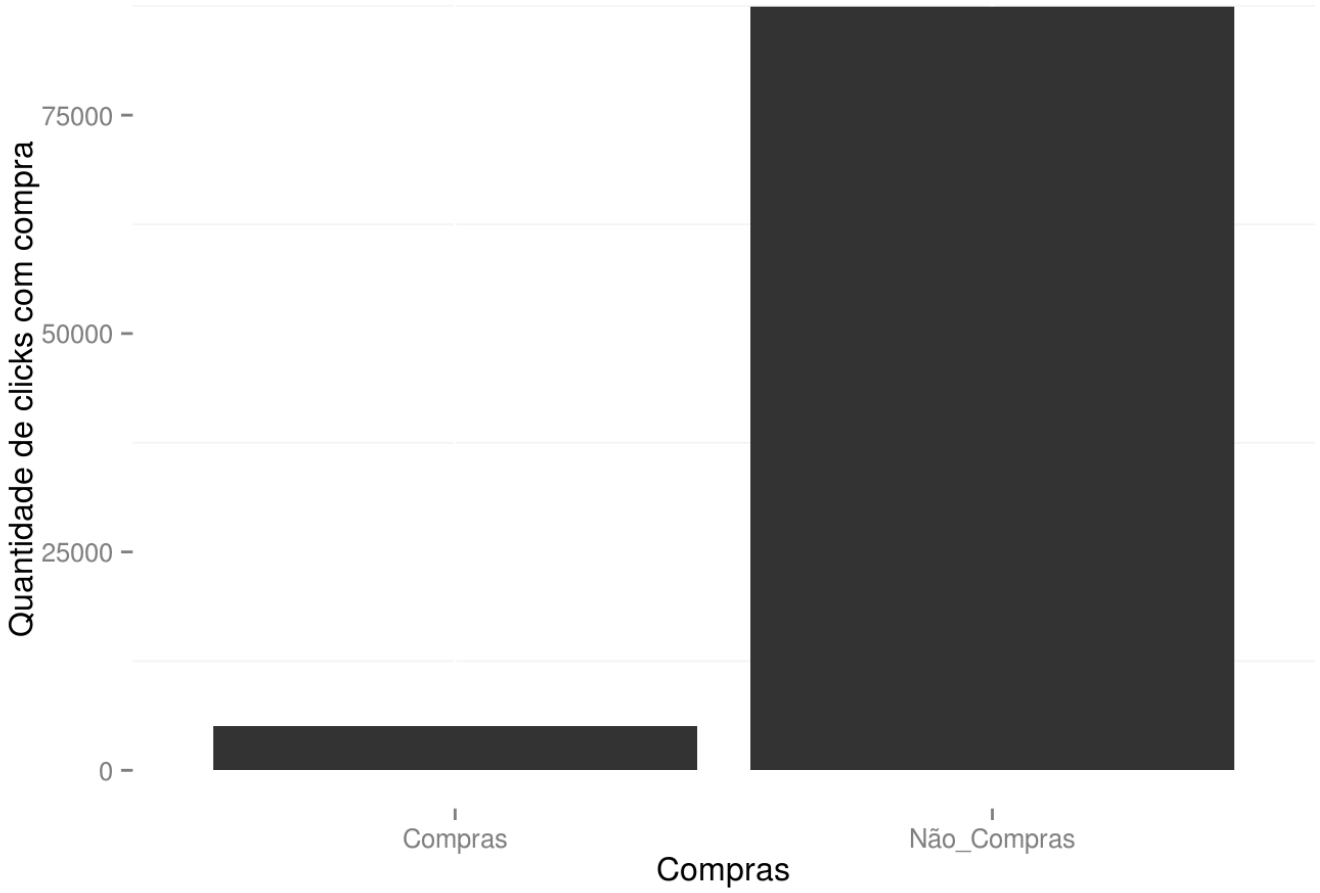
Como o dataset é bastante grande, para esse experimento vamos utilizar apenas 0.1% do dataset. Essa amostra foi retirada de forma aleatória.

```
require(ggplot2)
require(dplyr)

recSys<- read.csv("~/Projetos/DataAnalysis/Assignment4/RecSys.csv")

colnames(recSys) <- c("SESSION", "DAY", "MONTH", "TIME", "ITEM", "CATEGORY", "W
EEKDAY", "CLICKED", "BOUGHT", "SOLDABILITY", "SAME_CAT", "SOLD_MEAN", "SOLD_MEA
N_DIFF", "SOLD_MEDIAN", "SESSION_SIZE", "CATEG_MOST", "IS_BUY","SESSION_DURATIO
N", "RELATIVE_TIME", "RELATIVE_TIME_PROP")

recSys$SESSION <- as.factor(recSys$SESSION)
recSys$MONTH <- as.factor(recSys$MONTH)
recSys$ITEM <- as.factor(recSys$ITEM)
recSys$IS_BUY <- as.factor(recSys$IS_BUY)

summary(recSys)
```

```
##      SESSION           DAY        MONTH          TIME
##  11420919:  124   Min.   : 1.00   4:16661   18:q4  : 1923
##  6902187 :   99   1st Qu.: 9.00   5:15341   18:q3  : 1918
##  6159198 :   83   Median :15.00   6:14505   19:q4  : 1899
##  10838496:   79   Mean   :15.69   7:12336   19:q1  : 1804
##  1227469 :   69   3rd Qu.:23.00   8:18371   19:q2  : 1738
##  2419503 :   64   Max.   :31.00   9:15260   20:q1  : 1711
##  (Other) :91956                             (Other):81481
##         ITEM          CATEGORY     WEEKDAY        CLICKED
##  643078800:  399   0      :45955   FRI:10280   Min.   :     1
##  214829878:  366   S      :29855   MON:18019   1st Qu.:  2297
##  214853094:  323   1      : 4670   SAT:11242   Median :  7676
##  214826610:  212   2      : 3586   SUN:20500   Mean   : 13239
##  214853420:  207   3      : 2366   THU:12831   3rd Qu.: 16823
##  214853096:  186   4      : 1515   TUE: 6176   Max.   :147419
##  (Other)  :90781   (Other): 4527   WED:13426
##      BOUGHT         SOLDABILITY         SAME_CAT         SOLD_MEAN
##  Min.   :    0.0   Min.   :  0.000   Min.   :  1.000   Min.   :  0.000
##  1st Qu.:    7.0   1st Qu.:  0.220   1st Qu.:  2.000   1st Qu.:  0.470
##  Median :   52.0   Median :  0.920   Median :  4.000   Median :  1.210
##  Mean   :  336.5   Mean   :  2.261   Mean   :  6.693   Mean   :  2.258
##  3rd Qu.:  272.0   3rd Qu.:  2.800   3rd Qu.:  7.000   3rd Qu.:  2.890
##  Max.   :10226.0   Max.   :121.580   Max.   :116.000   Max.   :106.730
##
##  SOLD_MEAN_DIFF       SOLD_MEDIAN       SESSION_SIZE       CATEG_MOST
##  Min.   :-48.07000   Min.   :  0.000   Min.   :  1.000   S      :46682
##  1st Qu.: -0.53000   1st Qu.:  0.320   1st Qu.:  2.000   2      :12488
##  Median :  0.00000   Median :  0.940   Median :  4.000   1      :10460
##  Mean   : -0.00188   Mean   :  2.213   Mean   :  7.763   3      : 6641
##  3rd Qu.:  0.29000   3rd Qu.:  2.560   3rd Qu.:  9.000   4      : 4067
##  Max.   : 90.61000   Max.   :106.730   Max.   :124.000   5      : 2934
##                                                          (Other): 9202
##  IS_BUY    SESSION_DURATION  RELATIVE_TIME     RELATIVE_TIME_PROP
##  0:87451   Min.   :    0.0   Min.   :    0.0   Min.   :0.0000
##  1: 5023   1st Qu.:  107.3   1st Qu.:    0.0   1st Qu.:0.0000
##            Median :  335.8   Median :  105.1   Median :0.4600
##            Mean   :  801.8   Mean   :  392.4   Mean   :0.4811
##            3rd Qu.:  919.3   3rd Qu.:  391.3   3rd Qu.:0.9800
##            Max.   :17964.3   Max.   :17964.3   Max.   :1.0000
##
```

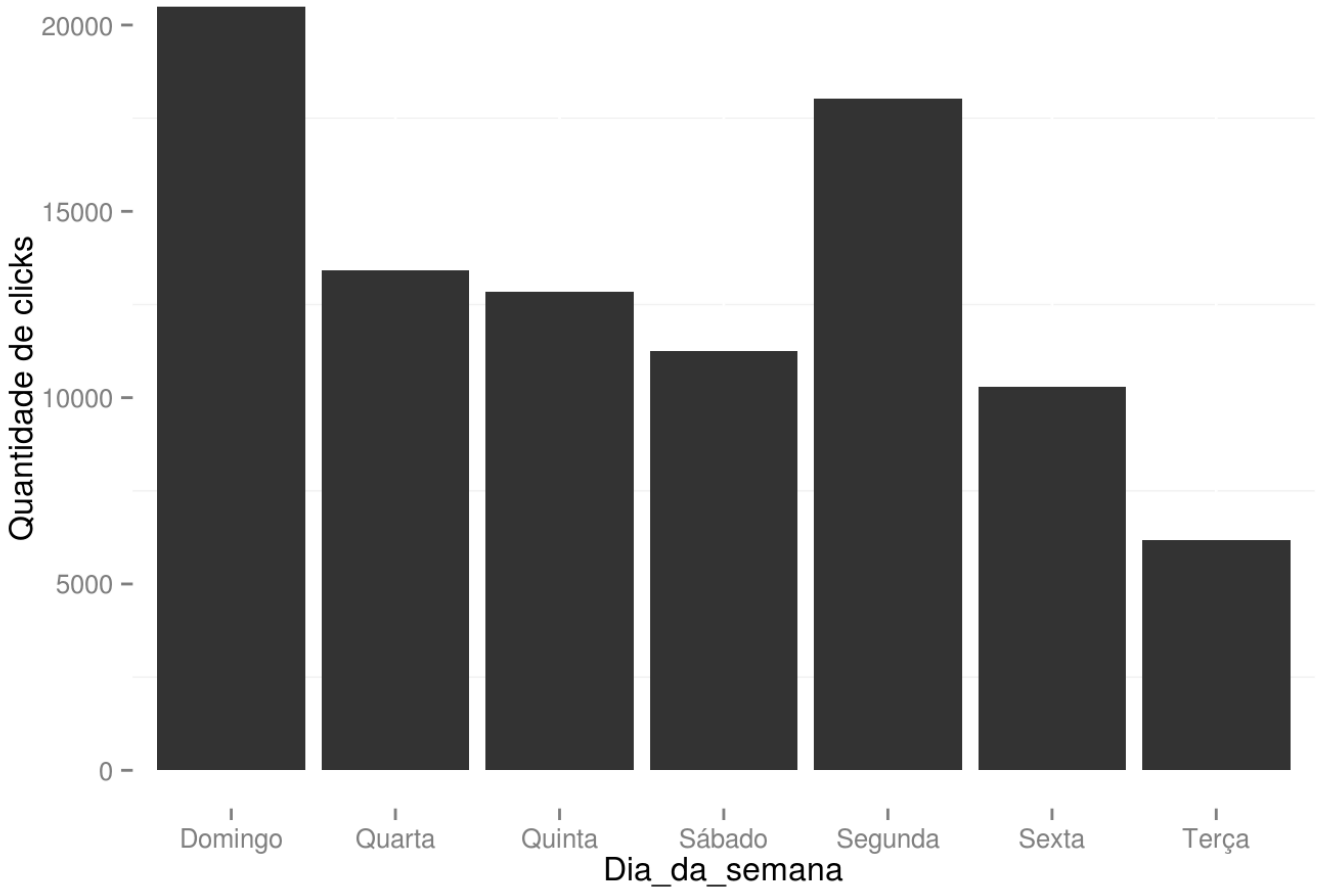
Olhando apenas para os dados é possível tirar algumas conclusões:

```
isBuy <- as.data.frame(summary(recSys$IS_BUY))

isBuy["Compras"] <- c("Não_Compras", "Compras")
colnames(isBuy) <- c("Quantidade", "Compras")

ggplot(isBuy, aes(x=Compras, y=Quantidade)) +
  geom_bar(stat="identity") +
  labs(y='Quantidade de clicks com compra') +
  theme(panel.background=element_blank())
```



Existem muito mais clicks com não compra do que com compras. Mostrando que os dados estão desbalanceados. 5.5% são os clicks com compras e o restante para clicks com não compras. Além disso, podemos obervar uma diferença no volumes de clicks ao longo dos dias da semana:

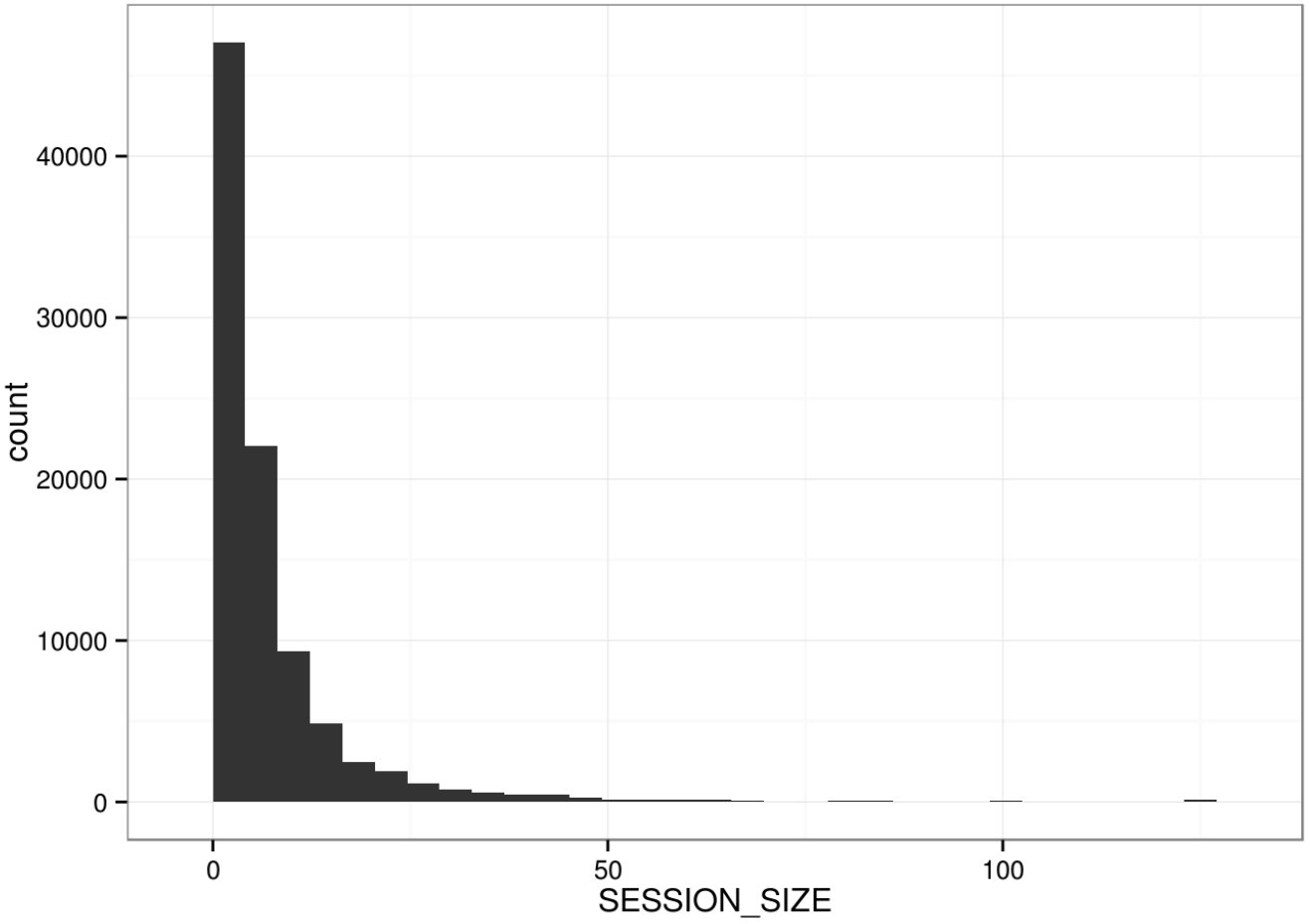
```
weekday <- as.data.frame(summary(recSys$WEEKDAY))

weekday["Dia_da_semana"] <- c("Sexta", "Segunda", "Sábado", "Domingo", "Quint
a", "Terça", "Quarta")
colnames(weekday) <- c("Quantidade", "Dia_da_semana")

ggplot(weekday, aes(x=Dia_da_semana, y=Quantidade)) +
  geom_bar(stat="identity") +
  labs(y='Quantidade de clicks') +
  theme(panel.background=element_blank())
```



Nas segundas e nos domingos é possível identificar um maior número de clicks em relação aos outros dias.

A grande maioria das sessões possui menos de 10 clicks. Como podemos observar no histograma abaixo:

```
ggplot(recSys, aes(SESSION_SIZE)) + geom_histogram() +  theme_bw()
```



Para melhor avaliar o modelo que será criado, precisamos antes realizar uma divisão nos dados da amostra. Reservando 1/3 da amostra para teste e 2/3 para treino.

```
indexes = sample(1:nrow(recSys), size=0.3*nrow(recSys))

teste = recSys[indexes,]
treino = recSys[-indexes,]

recSys <- NULL
```

Para o nosso primeiro modelo, vamos inicialmente observar o comportamento do classificador utilizando apenas o dia da semana.

```
bm <- glm(IS_BUY ~ WEEKDAY,
          data = treino,
          family = "binomial")

summary(bm)
```

```
##
## Call:
## glm(formula = IS_BUY ~ WEEKDAY, family = "binomial", data = treino)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.3883  -0.3730  -0.3217  -0.2834   2.6058
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.956489   0.054360 -54.387  < 2e-16 ***
## WEEKDAYMON   0.021206   0.067808   0.313   0.7545
## WEEKDAYSAT   0.409514   0.069619   5.882 4.05e-09 ***
## WEEKDAYSUN   0.326194   0.063829   5.110 3.21e-07 ***
## WEEKDAYTHU  -0.238325   0.076729  -3.106   0.0019 **
## WEEKDAYTUE  -0.404561   0.100700  -4.017 5.88e-05 ***
## WEEKDAYWED  -0.006796   0.072328  -0.094   0.9251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27150  on 64731  degrees of freedom
## Residual deviance: 26963  on 64725  degrees of freedom
## AIC: 26977
##
## Number of Fisher Scoring iterations: 6
```

```
exp(bm$coefficients)
```

```
## (Intercept)  WEEKDAYMON  WEEKDAYSAT  WEEKDAYSUN  WEEKDAYTHU  WEEKDAYTUE
##  0.05200117  1.02143239  1.50608576  1.38568356  0.78794664  0.66726953
##  WEEKDAYWED
##  0.99322702
```

É possível identificar que em relação a Sexta-Feira, o Sábado e o Domingo são os dias que mais tem chance de ocorrer uma compra. O sábado tem 1.418 mais chances de ocorrer uma compra do que em uma sexta. Já o domingo tem 1.321 mais chances.

Para testar o nosso classificador vamos utilizar os dados de teste.

```
predictions <- predict(bm, type = "response", newdata = teste) > 0.07
verdadeiras_compras <- teste$IS_BUY == 1

table(predictions, verdadeiras_compras)
```

```
##            verdadeiras_compras
## predictions FALSE  TRUE
##       FALSE 23049  1299
##       TRUE   3156   238
```

Existem dois tipos de erros que podemos observar na tabela acima. O primeiro erro é o falso positivo, que ocorre quando o modelo prediz que uma saída é verdadeira quando na verdade ela é possitiva. Para o nosso caso é o mesmo que dizer que o modelo achou que um click resultava em compra quando na verdade ele resultava em não compra.

O segundo erro é o falso negativo, que ocorre quando o modelo prediz que uma saída é falsa quando na verdade ela é possitiva. Para o nosso caso é o mesmo que dizer que o modelo achou que um click não era compra quando na verade ele resultava em compra.

O nosso objetivo é diminuir ao máximo o segundo tipo de erro. Esse modelo testado errou 84% do clicks de compra. Acertando apenas 26% do clicks de compras. Sendo esse um valor muito baixo.

Em busca de melhorar esse valor, realizamos um novo teste. Dessa vez modificando o limiar para 0.05

```
predictions <- predict(bm, type = "response", newdata = teste) > 0.05
verdadeiras_compras <- teste$IS_BUY == 1

table(predictions, verdadeiras_compras)
```

```
##            verdadeiras_compras
## predictions FALSE  TRUE
##       FALSE 12183   608
##       TRUE  14022   929
```

Nesse modelo temos um erro de 34% dos clicks de compra. O modelo deveria ter classificado como compra e classificou como não compra.

Em busca de melhorar esse número criamos outro modelo. Dessa vez considerando o máximo de atributos possivel. Devido ao tamanho do dataset decidimos focar em alguns atributos para esse relatório. Os atributos escolhidas foram: SOLDABILITY, WEEKDAY, MONTH, RELATIVE_TIME_PROP, RELATIVE_TIME, SESSION_DURATION, SESSION_SIZE, DAY, BOUGHT.

```
bm2 <- glm(IS_BUY ~ SOLDABILITY + WEEKDAY + MONTH + RELATIVE_TIME_PROP + RELATI
VE_TIME + SESSION_DURATION + SESSION_SIZE + DAY + BOUGHT,
          data = teste,
          family = "binomial")

summary(bm2)
```

```
##
## Call:
## glm(formula = IS_BUY ~ SOLDABILITY + WEEKDAY + MONTH + RELATIVE_TIME_PROP +
##     RELATIVE_TIME + SESSION_DURATION + SESSION_SIZE + DAY + BOUGHT,
##     family = "binomial", data = teste)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7800  -0.3484  -0.3043  -0.2667   2.7965
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.782e+00  1.119e-01 -24.851  < 2e-16 ***
## SOLDABILITY         6.182e-02  4.877e-03  12.675  < 2e-16 ***
## WEEKDAYMON         -1.508e-01  1.023e-01  -1.474 0.140478
## WEEKDAYSAT          1.379e-01  1.050e-01   1.313 0.189095
## WEEKDAYSUN          1.786e-01  9.410e-02   1.897 0.057768 .
## WEEKDAYTHU         -4.641e-01  1.166e-01  -3.982 6.85e-05 ***
## WEEKDAYTUE         -4.008e-01  1.424e-01  -2.815 0.004885 **
## WEEKDAYWED         -1.261e-01  1.071e-01  -1.178 0.238700
## MONTH5             -2.317e-01  8.982e-02  -2.580 0.009885 **
## MONTH6             -4.711e-01  9.805e-02  -4.805 1.55e-06 ***
## MONTH7             -1.543e-01  9.391e-02  -1.643 0.100424
## MONTH8             -2.917e-01  8.394e-02  -3.475 0.000511 ***
## MONTH9             -3.914e-01  8.774e-02  -4.462 8.14e-06 ***
## RELATIVE_TIME_PROP -1.453e-01  7.612e-02  -1.909 0.056263 .
## RELATIVE_TIME       3.820e-05  4.167e-05   0.917 0.359281
## SESSION_DURATION    1.396e-04  2.559e-05   5.455 4.91e-08 ***
## SESSION_SIZE        9.050e-03  2.363e-03   3.830 0.000128 ***
## DAY                -8.173e-03  3.139e-03  -2.604 0.009215 **
## BOUGHT              7.889e-05  2.344e-05   3.365 0.000764 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11881  on 27741  degrees of freedom
## Residual deviance: 11377  on 27723  degrees of freedom
## AIC: 11415
##
## Number of Fisher Scoring iterations: 6
```

Como os atributos RELATIVE_TIME_PROP, RELATIVE_TIME e DAY se mostraram pouco relevantes, decidimos por tirar esses atributos.

```
bm2 <- glm(IS_BUY ~ SOLDABILITY + WEEKDAY + MONTH + SESSION_DURATION + SESSIO
N_SIZE + BOUGHT,
          data = teste,
          family = "binomial")
summary(bm2)
```

```
##
## Call:
## glm(formula = IS_BUY ~ SOLDABILITY + WEEKDAY + MONTH + SESSION_DURATION +
##     SESSION_SIZE + BOUGHT, family = "binomial", data = teste)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7155  -0.3473  -0.3054  -0.2696   2.7991
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.970e+00  9.631e-02 -30.834  < 2e-16 ***
## SOLDABILITY       6.145e-02  4.856e-03  12.656  < 2e-16 ***
## WEEKDAYMON       -1.537e-01  1.023e-01  -1.503 0.132940
## WEEKDAYSAT        1.303e-01  1.049e-01   1.242 0.214126
## WEEKDAYSUN        1.688e-01  9.403e-02   1.795 0.072668 .
## WEEKDAYTHU       -4.699e-01  1.165e-01  -4.032 5.54e-05 ***
## WEEKDAYTUE       -3.965e-01  1.423e-01  -2.786 0.005337 **
## WEEKDAYWED       -1.333e-01  1.070e-01  -1.247 0.212570
## MONTH5           -2.277e-01  8.973e-02  -2.537 0.011178 *
## MONTH6           -4.666e-01  9.797e-02  -4.762 1.91e-06 ***
## MONTH7           -1.695e-01  9.358e-02  -1.812 0.070035 .
## MONTH8           -3.059e-01  8.375e-02  -3.653 0.000260 ***
## MONTH9           -3.798e-01  8.762e-02  -4.335 1.46e-05 ***
## SESSION_DURATION  1.602e-04  1.728e-05   9.274  < 2e-16 ***
## SESSION_SIZE      8.900e-03  2.344e-03   3.797 0.000147 ***
## BOUGHT            7.820e-05  2.344e-05   3.336 0.000848 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11881  on 27741  degrees of freedom
## Residual deviance: 11387  on 27726  degrees of freedom
## AIC: 11419
##
## Number of Fisher Scoring iterations: 6
```

```
predictions <- predict(bm2, type = "response", newdata = teste) > 0.05
verdadeiras_compras <- teste$IS_BUY == 1

table(predictions, verdadeiras_compras)
```

```
##            verdadeiras_compras
## predictions FALSE  TRUE
##       FALSE 15200   508
##       TRUE  11005  1029
```

Para esse novo modelo testado temos um erro de 31% do clicks de compra. Acertando 69% do clicks de compras. Sendo esse um valor muito mais alto em comparação ao nosso primeiro modelo.

Nos próximos passos temos que derivar mais atributos para assim diminuir o valor dos falsos positivos.