Cliques em Propagandas que são Convertidos em Download

Guilherme Maia Batista

1 Entendendo o Problema de Negócio

O sucesso de uma empresa está associado não apenas à qualidade do seu produto/serviço, como também à sua capacidade de alcançar potenciais clientes. Por isso a importância de investimentos na área de publicidade.

Divulgar um produto por meio de anúncios na internet é uma das formas mais efetivas de dar-lhe visibilidade, sendo que os custos desses anúncios estão associados ao número de cliques que recebem. Dessa forma, é essencial identificar quais cliques possuem maior chance de serem convertidos em download do aplicativo.

Para melhorar a capacidade de identificação de cliques convertidos em download, a empresa TalkingData disponibilizou um dataset [1] com informações sobre diversos cliques rotulados como 0 (não convertido) ou 1 (convertido). O objetivo deste estudo é, a partir do dataset disponibilizado, construir um modelo que classifique com precisão um clique como convertido ou não em download.

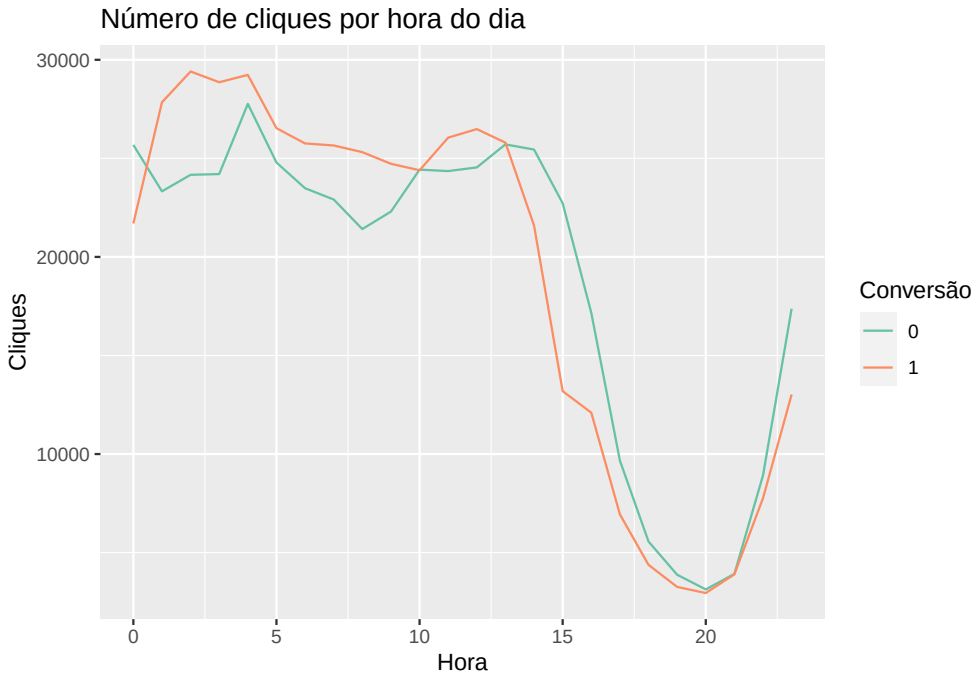
2 Entendendo os Dados

O dataset é composto de 184.903.890 cliques realizados durante o período de 4 dias. Porém, apenas 456.846 cliques foram convertidos em download, menos de 1%. Esse desbalanceamento de classes pode causar problemas durante o treinamento de um modelo: ele aprenderia muito sobre os cliques não convertidos e quase nada sobre os cliques convertidos.

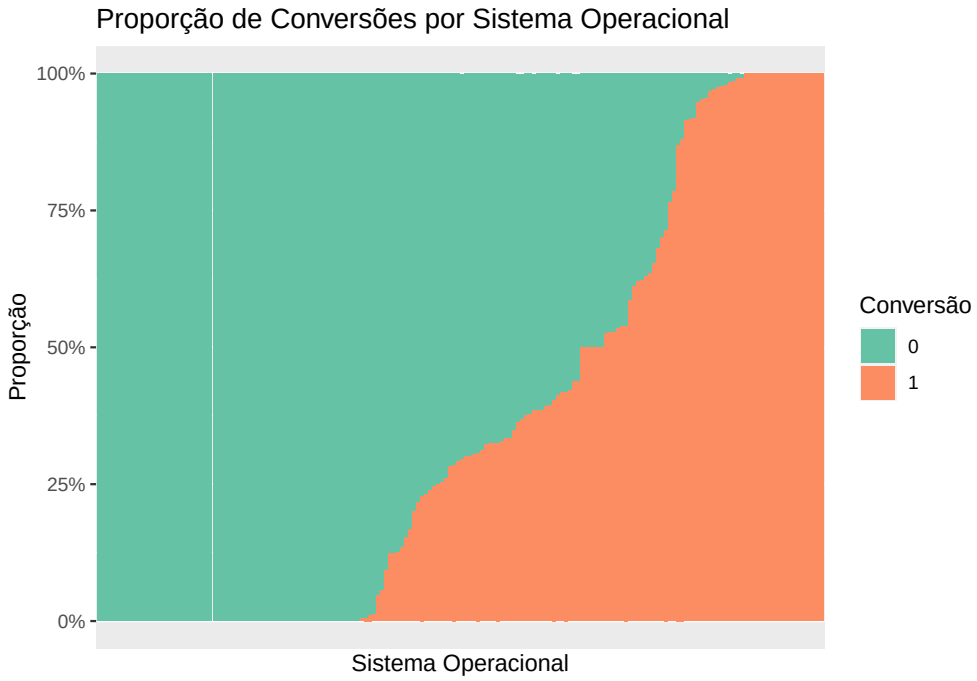
Para contornar isso, foi utilizada a técnica de undersampling, que consiste em extrair apenas uma amostra da classe majoritária para que o dataset tenha uma proporção balanceada entre as classes. Assim, o dataset considerado para o restante da análise consistiu em 913.693 observações (metade para cada categoria). Na tabela abaixo está a descrição de cada variável.

nome_da_variavel	tipo	descricao
ip	categórica nominal	Endereço IP do clique
app	categórica nominal	ID do aplicativo que mostrou o anúncio
device	categórica nominal	Modelo do celular
os	categórica nominal	Sistema operacional do celular
channel	categórica nominal	ID do canal do publicador do anúncio
click_time	data e hora	Data e hora em que o clique foi efetuado
attributed_time	data e hora	Se o usuário efetuou o download, mostra a data e hora do download
is_attributed	categórica nominal	Variável target, indica se o clique é convertido ou não

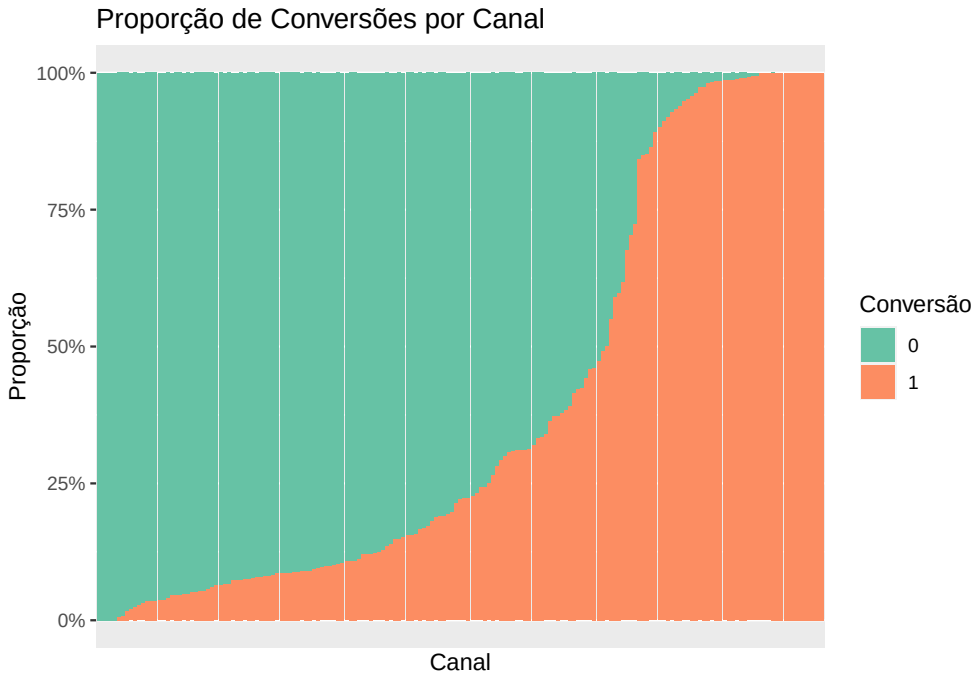
[1] https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data



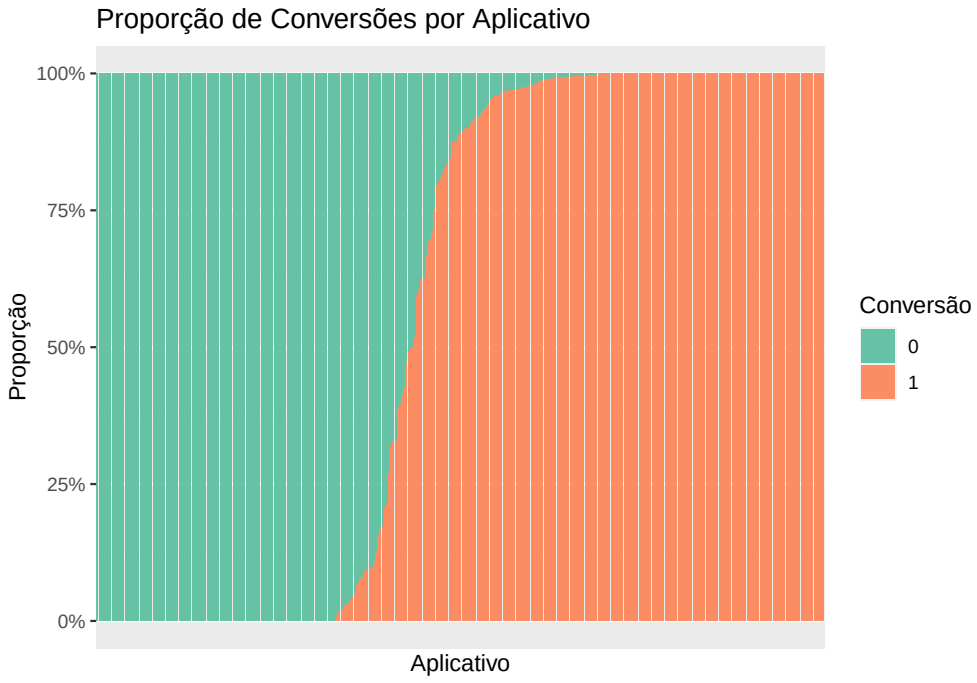
O comportamento dos cliques durante as horas do dia parecem seguir o mesmo padrão.



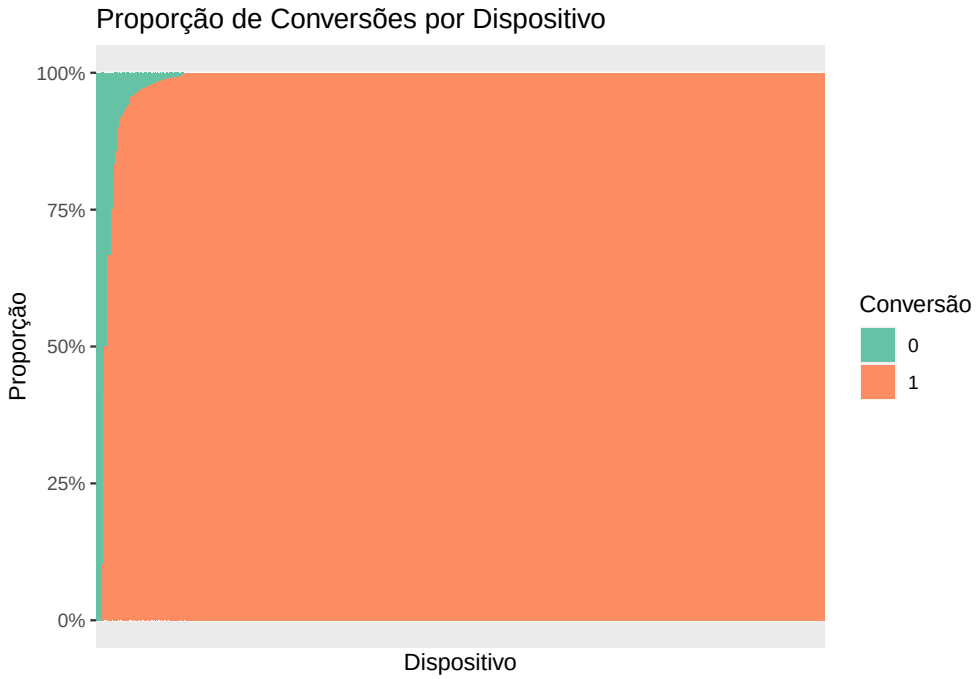
Nesse gráfico cada coluna representa um sistema operacional diferente. Nele é possível observar que há uma quantidade maior de sistemas operacionais sem nenhum clique convertido em download do que sistemas operacionais com apenas cliques convertidos.



A maioria dos canais tem na sua composição menos de 50% de cliques convertidos. Mas alguns canais possuem praticamente apenas cliques convertidos.



Para os aplicativos para que há uma variação polarizada: ou o aplicativo possui aproximadamente 100% de cliques não convertidos ou ele possui aproximadamente 100% de cliques convertidos.



Esse gráfico chama bastante atenção, pois, dentre os 1883 tipos diferentes de dispositivos, quase todos apresentam apenas cliques convertidos em download.

Construção do Modelo

Antes de construir o modelo, a base foi separada em duas de forma aleatória. 90% da base foi utilizada para treinar o modelo e 10% para avaliar a qualidade do modelo (geralmente divide-se a base em 60% para treino e 40% para teste, mas como 10% já representaria 90 mil observações, considerou-se essa divisão). As variáveis click_time e attributed time foram desconsideradas, mas a partir da variável click_time criou-se a variável click_hour (hora que o clique foi feito), que foi testada nos modelos propostos.

O algoritmo escolhido foi o classificador Naive Bayes. Algumas combinações de variáveis foram testadas, além dos parâmetros do modelo. A melhor combinação utilizou todas as variáveis, exceto a clicK_hour. Abaixo está a matrix de confusão.

```
mod_final = naiveBayes(is_attributed~.-click_hour, data = df_train, laplace = 2)

res = modResult(mod_final)

kable(res[[1]])
```

	0	1
0	44772	845
1	7539	38214

A precisão do modelo final foi de 90,82% de acerto.