Leveraging Snowplow Event Tracking for Better Understanding Visitor Conversion

Benjamin S. Knight

January 18, 2017

1 Background

The central goal of this research is marketing analytics. Snowplow is a web event tracker capable of handling tens of millions of events per day. Using this data, we hope to answer the question of how different visitor experiences at a company's marketing site relate to the probability of those visitors ultimately becoming paying customers.

Applying machine learning to event data gathered from web applications is becoming standard practice.[1] However, here we are trying to make inferences about customers and potential customers using information attained from before they first used the application. In fact, the overwhelming majority of observations come from users who have never used the application at all.

The Snowplow data we have at our disposal can be thought of as a much larger, richer version of the MSNBC.com Anonymous Web Data Data Set hosted by the University of California, Irvine's Machine Learning Repository.[2] Like the MSNBC.com dataset, we have access to page view events along with other kinds of events. Unlike the MSNBC.com data set, we can map these events to individual cookies, then map the cookies to accounts of application users.

This sort of information can be useful in determining what portions of the marketing site should receive priority with respect to A/B testing, new content, and so forth. At the same time, having awareness of visitors with a higher than normal likelihood of becoming customers would help the Sales Department better utilize scarce resources.

2 Problem Statement

To what extent can we infer a visitor's likelihood of becoming a paying customer based upon that visitor's experience on the company marketing site? Assuming that the predicted visitor behavior is superior to blind guessing, what specific factors (both within and outside of the company's control) contribute to a visitor's likelihood of becoming a paying customer?

In more concrete terms, we are essentially confronted with a binary classification problem - will the account in question add a credit card (cc_date_added IS NOT NULL 'yes'/'no')? This labeling information is contained in the 'cc' column within the file 'munged_df.csv.'

3 Datasets and Inputs

The raw Snowplow data available to us is 15 gigabytes spanning 300 variables and tens of millions of events from November 2015 to January 2017. When we omit fields that are not in active use, are redundant, contain personal identifiable information (P.I.I.), or which cannot have any conceivable baring on customer conversion, then we are left with 14.6 million events and the variables shown in Table 1.

I use the phrase 'variable' as opposed to feature, since this dataset will need to undergo substantial

[1] Falchuk, Mesterharm, and Panagos. 2016. (http://www.research.rutgers.edu/~mesterha/emerging_2016_3_40_50024.pdf)

[2] Heckerman, David(https://archive.ics.uci.edu/ml/datasets/MSNBC.com+Anonymous+Web+Data)

transformation before we can employ any supervised learning technique. Each row has an 'event_id' along with an 'event_name' and a 'page_url.' The event_id is the row's unique identifier, the event_name is the type of event, and the page_url is the URL within the marketing site where the event took place.

In transforming the data, we will need to create features by creating combinations of event types and distinct URLs, and counting the number of occurrences while grouping on accounts. For instance, if '.../payment_plan.com' is a frequent page url, then the number of page views on payment_plan.com would be one feature, the number of page pings would be another, as would the number of web forms submitted, and so forth. Given that there are six distinct event types and dozens of URLs within the marketing site, then the feature space will likely be in the hundreds of features. This feature space will only widen as we add additional variables to the mix including geo_region, number of visitors per account, and so forth.

At the same time, we will need to filter the data. Since we are interested in the causal relationship between visitors' marketing site experiences and whether they ultimately became paying customers, we can and should omit all events that occur after the time-stamp 'cc_date_added' - the date when a customer first added a credit card to their account.

Snowplow Variable Name	Snowplow Variable Description
event_id	The unique Snowplow event identifier
account_id	The account number if an account is associated with the domain_userid
reg_date	The date an account was registered
cc_date_added	The date a credit card was added
collector_tstamp	The timestamp (in UTC) when the Snowplow collector first recorded the event
domain_userid	This corresponds to a Snowplow cookie and will tend to correspond to a single internet device
domain_sessionidx	The number of sessions to date that the domain_userid has been tracked
domain_sessionid	The unique identifier for the Snowplow cookie/session
event_name	The type of event recorded
geo_country	The ISO 3166-1 code for the country that the visitor's IP address is located
geo_region_name	The ISO-3166-2 code for country region that the visitor's IP address is in
geo_city	The city the visitor's IP address is in
page_url	The page URL
page_referrer	The URL of the referrer (previous page)
mkt_medium	The type of traffic source (e.g. 'cpc', 'affiliate', 'organic', 'social')
mkt_source	The company / website where the traffic came from (e.g. 'Google', 'Facebook'
se_category	The event type
se_action	The action performed / event name (e.g. 'add-to-basket', 'play-video')
br_name	The name of the visitor's browser
os_name	The name of the vistor's operating system
os_timezone	The client's operating system timezone
dvce_ismobile	Is the device mobile? (1 = 'yes')

Table 1: Snowplow Variables within the Available Data

4 Solution Statement

I propose a two-part solution to this problem. The first question is how reliably can we predict conversion from visitor to paying customer. To this end, I will first experiment with SVMs using a RBF kernel. Other approaches may ulrimately prove to be more successful, but a RBF kernel is likley to serve us well as a starting point [3]

The second question we must bear in mind is whether the results are actionable. Specifically, we want to know what features are most relevant for customer conversion. For instance, does visiting the pricing page increase or decrease the likelihood of the visitor becoming a paying customer? If so, how strong or weak is the correlation? To this end we can use logistic regression. Logistic regression has the advantage of being one of the most readily interpretable machine learning techniques. Specifically, we can transform the resulting coefficients into their log-likelihood equivalents and insight into the likely effect size and its statistical significance.

5 Benchmark Model

For a *very* rough baseline of our future model's performance, we can divide the number of distinct timestamps where a credit card was added (cc_date_added) by the number of distinct Snowplow cookies (domain_userid). In this fashion, 4,298/1,014,324 = 0.004. On average, it takes 250 unique visitors to generate 1 paying customer. Put in different terms, we can predict with 0.996% accuracy that all visitors will fail to convert to paying customers. However, this finding is not particularly helpful - thus the need for additional evaluation metrics.

6 Evaluation Metrics

For assessing the efficacy of the machine learning model, I will employ a standard confusion matrix. While the confusion matrix will offer some important context, the ultimate evaluation metric will be the area under a precision-recall curve (AUC). Given that positive conversion events are extremely rare (0.4%), a precision-recall curve is more appropriate in this context compared to the Receiver Operator Characteristic (ROC) curve.[4]

For the subsequent logistic model, it will be necessary to omit as many features from the model as possible (we are likely interested in the top 20 or so largest, statistically significant coefficients). Pruning features from the resulting model will likely be a process of trial and error. To this end, we can select the best performing model using the Bayesian Information Criterion (BIC).

[3] https://arxiv.org/pdf/1606.00930v1.pdf

[4] http://ftp.cs.wisc.edu/machine-learning/shavlik-group/davis.icml06.pdf