Leveraging Snowplow Event Tracking for Better Understanding Visitor Conversion

Benjamin S. Knight

January 22, 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? 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 approximately 15 gigabytes spanning over 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 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.

[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`)

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 Pre-Transformation

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.

The transformed data set is viewable in the file 'munged_df.csv' and is approximately 38 MB. It contains 16,607 observations and a total of 708 features - 290 of which represent counts of various combinations of web events and URLs grouped by account_id. Next there are two aggregated features - the total number of distinct cookies associated with the account and the sum total of all Internet sessions linked to that account.

Within the feature space there are 151 features that represent counts of page view events linked to IP addresses within a certain country (e.g. a count of page views from China, a count of page views from France, and so forth).

Next, there are 46 features that represent counts of page views coming from a specific marketing medium ('mkt_medium'). Recall that 'mkt_medium' is the type of traffic. Examples include 'partner_link,' 'adroll,' or 'appstore.' The 'mkt_medium' subset of features is followed by 86 features that correspond to Snowplow's 'mkt_source' field. 'mkt_source' designates the company / website where the traffic came from. Examples from this subset of the feature space include counts of page views from Google.com ('mkt_source_google_com') and Squarespace ('mkt_source_square').

There are two additional feature: 'mobile_pageviews' and 'non_mobile_pageviews' that represent counts of page views that took place on mobile versus non mobile devices. I have also included an additional feature derived from these two - the share of page views that took place on a mobile device.

The final two types of features include counts of page views on a given browser (80 features) and counts of page views on a given operating system (46). The labels are in the last column and take a value of '1' (added a credit card) or '0' (did not add a credit card).

Regarding the allocation of data for testing and training purpose, we are not particularly constrained regarding the extent of the available data. I will use an .8 /.2 split of the data for training and testing purposes respectively.

4 Solution Statement

How reliably can we predict conversion from visitor to paying customer? In other words, how precisely can we predict whether the 'cc' label is a 0 or 1? For a binary classification problem of this sort, Support Vector Machines (SVMs) with a RBF kernel is likely to serve well as a starting point [3]. We have a large set of features (708) with plenty of observations (16,607) and so a linear SVM with no kernel may also be worth exploring.

5 Benchmark Model

For a *very* rough baseline of our future model's performance, we can divide the number of accounts where credit card was added (cc_date_added) by the number of total accounts. Thus, for our highly imbalanced data the probability of a label denoting a successful customer conversion (cc = 1) is 0.06. The imbalanced nature of the data means that accuracy is not a particularly helpful metric. Instead, I use the Area Under the Curve (AUC) of a Precision-Recall plot. Applying the data post-transformation (see munged_df.csv) directly to a SVM with RBF kernel and all default settings yielded an AUC of 0.22 - the benchmark model

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

for this project. The goal of this project is to see if, via hyper-parameter tuning, we can exceed an AUC of .3.

6 Evaluation Metrics

Given that positive conversion events are extremely rare (6%), a precision-recall curve is more appropriate in this context compared to the Receiver Operator Characteristic (ROC) curve.[4] Recall that a conversion to a paying account is represented by a non-NULL cc_date_added - a date field that is transformed into the boolean "CC.' The formulas for precision and recall are shown below.

$$precison = \frac{|\text{Paying Accounts Identified} \cap \text{Accounts Identified}|}{\text{Accounts Identified}}$$

$$recall = \frac{|\text{Paying Accounts Identified} \cap \text{Accounts Identified}|}{\text{Paying Accounts Identified}}$$

7 Work Flow

The first stage of the project will involve transforming the data set from millions of events into a data set of accounts with counts of events as features (see Datasets and Inputs). This is handled in 'Notebook 1 - Data Munging.' While the original raw data set is too vast to include, users can run Notebook 1 using the file 'raw_dataset_sample.csv' to see how the data transformation is handled.

The next stage is exploring the newly derived feature space. This is the purpose of 'Notebook 2 - Exploratory Analysis.' The notebook creates images with the purpose of visualizing the summary statistics of the feature set, and stores them in the 'Images' directory.

The next stage is creating a baseline model. This involves the implementation of a SVM with its default settings. 'Notebook 3 - Default SVM-RBF Model' establishes this baseline and plots the resulting precision-recall curve.

The next step involves tuning the hyper-parameters. To improve the likelihood of finding the most appropriate model model, I use two different approaches with their corresponding iPython notebooks - SVM with a RBF kernel ('Notebook 4 - RBF Kernel Hyper-Parameter Tuning') and linear SVM ('Notebook 5 - Linear SVC Hyper-Parameter Tuning').

The final stage will involve selecting the best model from either the SVM-RBF or linear SVM, implementing it, and writing up the subsequent results.

8 Alternate Approaches

This proposal has not emphasized dimensionality reduction, as exploratory analysis showed that use of Principal Component Analysis (PCA) typically corresponded to a several point reduction in the AUC. Depending on how the testing of SVM-RBF and linear SVM proceed, I may have recourse to experiment with linear discriminant analysis (LDA) since it incorporates information from the labels - a definite strength given the unbalanced nature of the data set.

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