Machine Learning Capstone Project

Domain Background

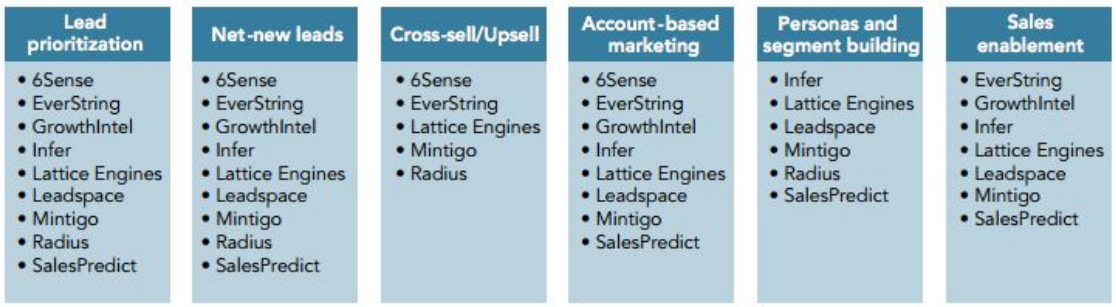
In recent years, technology has made it easier to bridge the gaps between siloed data sources, and provide a unified view of data across platforms, customer relationship management systems and marketing automation software. This enabled a new breed of products based on machine learning that aims to successfully predict whether a lead will convert into a customer.

Companies like Lattice Engines, Fliptop, 6sense and Infer are using machine learning to power sales and marketing teams, enabling them to quickly identify key characteristics of ideal target markets by uncovering purchasing patterns based on historical customer behavior and demographic data. In addition to historical data, some of them are using machine learning to analyze incoming data from external sources, like web searches and social media, to provide an even better understanding of who are the most promising customers.

As shown in the article "Probabilistic Modeling of a Sales Funnel to Prioritize Leads", by Brendan Duncan and Charles Elkan from the Department of Computer Science and Engineering at UCLA, from 2015, machine learning models can result in up to 307% increase in number of successful sales, as well as a dramatic increase in total revenue.

In fact, nearly two thirds of businesses have implemented predictive marketing analytics, including 42% that are expanding or upgrading their implementation within the next 12 months, according to Forresters From Insight to Action: How Predictive Analytics Improves B2B Marketing Outcomes, published in October 2015. Among businesses using predictive marketing analytics, 83% have experienced a positive business impact as a result of their implementation.

Selected predictive marketing analytics platform use cases



Many predictive marketing analytics vendors are rooted in lead scoring but have broadened their capabilities to include predictive modeling, personalization, and product recommendations that push deeper into the purchase funnel. However, none of them are built with a specific market segment in mind. Despite the fact that there is no established market leader among these players, and that the crowded field continues to attract new ventures, there is no specific solution tailored to the education market. This is our motivation.

Problem Statement

In June 2016 Udacity started to operate in Brazil. Since then, we have seen exponential growth in our Nanodegree students: average compounded weekly growth is 10-12% for the past 8 months. Our sales funnel is composed by the following stages:

1. Awareness: User discovered us through social network ad campaigns, PR, live events, or organically (SEO), and landed in our website;
2. Lead: User created a free account;
3. Marketing-qualified lead (MQL): User watched webinar(s), visited specific pages, opted-in our newsletter, followed Udacity in social network(s), and/or started a free course;
4. Sales-qualified lead (SQL): User visited our checkout more than once, made inquires about how our Nanodegree programs work, and/or tried to pay (unsuccessfully);
5. Student in trial: User started Nanodegree trial;
6. Paying student: User successfully concluded the trial and/or (directly) enrolled in a Nanodegree.

Although we currently use marketing automation, we are not using any user behaviour data to drive them toward the end of the funnel: we simply do promotional email campaigns to generate urgency to purchase. Usually, they are a one-size-fits-all campaigns targeted to MQLs. This raises the question:

> Given all behaviour data we collect from our users through the sales funnel, combined with additional data sources we may scrape from social networks, can we train an algorithm to successfully predict whether or not they are potential Udacity Nanodegrees students?

Datasets and Inputs

In order to tackle the proposed problem, we plan to use the datasets in the following table. Important to highlight: emails, names, or any other data that could identify users won't be shared in the project.

Dataset name	Description	Rows	Total fields	Field examples
auth_user	Contains key data about users	60,859	11	id, last_login, username, first_name, last_name, email, date_joined
payment_app_product	Contains key data about Nanodegrees	26	12	id, name, code, price
payment_app_sub	Contains all	5,760	39	id, access_until, cancel_requested, status, register_date, credit_card_retries,

scription	subscription data from paying students			chosen_payment_type_code, product_id, user_id, first_payment_date, cohort_id, full_amount, instalment_amount
frontend_brazil.pages	Contains data from all website visits	4,707,359	47	anonymous_id, category, context_ip, name, path, referrer, sent_at, timestamp, title, url, user_id, context_campaign_medium
frontend_brazil.identifies	Contains data that links visitors with user accounts	113,630	34	received_at, anonymous_id, user_id
frontend_brazil.tracks	Contains data from all events tracked in marketing website	1,092,373	31	received_at, event, event_text, timestamp, user_id, anonymous_id, context_ip
brazil_events.event_sign_up	Contains data from all webinars	26,931	54	email, enrollment_date, event, event_title, event_start_date, event_end_date, slug, user_id, user_interests, event_type
analytics_tables.course_enrollments	Contains data from all free course enrollments		15	user_id, course_key, join_time, leave_time, course_title, course_level
zendesk-data-2017-03-20-0132	Contains data from all	7,802	42	id, requester, requester id, requester email, submitter, assignee, group, subject, tags, status, priority, via, ticket type, created at, updated at, assigned at, organization, due date, initially assigned at, solved at

Solution Statement

Following the approach proposed in the article "Probabilistic Modeling of a Sales Funnel to Prioritize Leads", by Brendan Duncan and Charles Elkan from the Department of Computer Science and Engineering at UCLA, from 2015, the proposed solution is a DQM (direct qualification model). It models a sales funnel using a single multiclass classifier. Leads receive different class labels depending on how far along in the sales funnel they progress. We will classify leads into 4 classes:

- NoSUP: Leads that never sign-up for a free account.
- NoTRY: Leads that sign-up for a free account but never convert to trial.
- LOST: Leads that convert to trial that ultimately become Nanodegree students.
- WON: Leads that convert to trial that successfully become Nanodegree students.

Benchmark Model

Udacity US created a model to predict if a student will start a trial - equivalent to LOST probability. We will use that as benchmark, as it is a comparable approach used to the same purpose.

After extracting similar data from US data sources, the team from headquarters treated the data, divided into test/train set (⅓ of elements go to the train set) and ran the training set over the following algorithms: various support vectors machines with different parameters and kernels, stochastic gradient descent, linear, logistic regression.

They compared the results against accuracy (over the test dataset), where SVM and logistic gave approximately 72% and other ~64%. The highest value will be our benchmark.

Evaluation Metrics

We will use accuracy as evaluation metric: the fraction of correct predictions.

In multilabel classification, the function returns the subset accuracy. If the entire set of predicted labels for a sample strictly match with the true set of labels, then the subset accuracy is 1.0; otherwise it is 0.0.

If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the fraction of correct predictions over n_{samples} is defined as

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

where $1(x)$ is the indicator function.

Project Design

We plan to use all key supervised learning key methods, not only base estimators (e.g. Support Vector Machines, Stochastic Gradient Descent, Naive Bayes, Nearest Neighbors) but also ensemble methods (e.g. AdaBoost, Gradient Tree Boosting, VotingClassifier), and compare all results - using scikit-learn implementation.

Also, we plan to use 75% of the data for training and 25% of the data for testing. All behavioral data captured after trial period will be discarded to simplify the exercise.