Probabilistic Modeling of a Sales Funnel to Prioritize Leads

Brendan Duncan and Charles Elkan
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 90293-0404, USA
{baduncan, elkan}@cs.ucsd.edu

ABSTRACT

This paper shows how to learn probabilistic classifiers that model how sales prospects proceed through stages from first awareness to final success or failure.[1] Specifically, we present two models, called DQM for direct qualification model and FFM for full funnel model, that can be used to rank initial leads based on their probability of conversion to a sales opportunity, probability of successful sale, and/or expected revenue. Training uses the large amount of historical data collected by customer relationship management or marketing automation software. The trained models can replace traditional lead scoring systems, which are hand-tuned and therefore error-prone and not probabilistic. DQM and FFM are designed to overcome the selection bias caused by available data being based on a traditional lead scoring system. Experimental results are shown on real sales data from two companies. Features in the training data include demographic and behavioral information about each lead. For both companies, both methods achieve high AUC scores. For one company, they result in a a 307% increase in number of successful sales, as well as a dramatic increase in total revenue. In addition, we describe the results of the DQM method in actual use. These results show that the method has additional benefits that include decreased time needed to qualify leads, and decreased number of calls placed to schedule a product demo. The proposed methods find high-quality leads earlier in the sales process because they focus on features that measure the fit of potential customers with the product being sold, in addition to their behavior.

1. INTRODUCTION

Customer relationship management systems and marketing automation software have become popular tools for companies with sales and marketing teams. Because these systems store a large amount of historical sales data, they provide great potential for machine learning algorithms to improve the sales process. In theory, companies can use a predictive sales lead scoring or ranking model to prioritize sales and marketing efforts towards leads that are more likely to result in successful sales. This paper shows how to put this theory into practice.

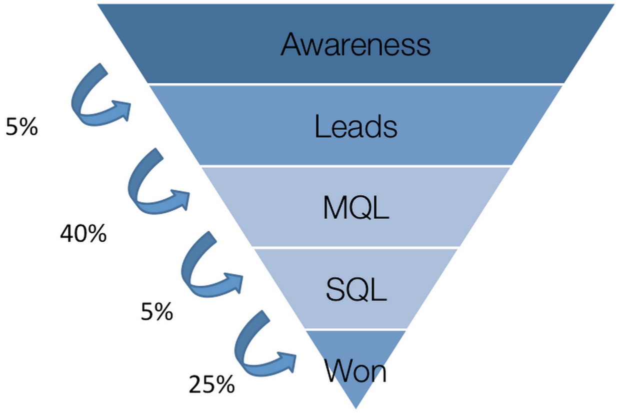

Figure 1: Sales funnel. MQL means marketing-qualified lead, while SQL means sales-qualified lead. Image copyright ©2015 by Fliptop, Inc.

Figure 1 shows a typical sales funnel. The different cross sections of the funnel represent different stages as a lead moves forward in the sales process, from the top of the funnel to the bottom. The decreasing diameter of the funnel represents a smaller and smaller volume of prospects reaching each successive stage.

1.1 Types of prospective customers

In Figure 1, a lead is an initial prospective customer that has not been evaluated in any way. For example, when an individual visits a website, or exchanges contact information with the marketing team, they will begin to be tracked by marketing automation software, as a "cold lead."

As leads are tracked by marketing teams, and by marketing automation software, marketing will qualify leads based on certain criteria, such as the amount of interest they show in the product (behavioral information) and their demographic fit for purchasing the product; see Section 1.3. Leads that are qualified by marketing will be passed along to the sales team and called "marketing-qualified leads."

Once the sales team receives leads from marketing, there is an additional qualification step. So-called teleprospectors or sales development representatives reach out to the individuals and determine if they meet the minimum criteria for becoming a sales opportunity. For example, the person must be in the market for the product or service offered by the seller, and must have the author-

[1] Research performed while Brendan Duncan worked at Fliptop Inc., 594 Howard Street, Suite 400, San Francisco, CA 94105.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788578 .

ity and budget to make a purchase within the sales timeline requirements. If an individual meets these criteria, he or she is qualified, called a “sales-qualified lead” (SQL), and becomes a genuine sales opportunity that is assigned to an account executive. This process is called “lead conversion.” The majority of SQLs will be pursued by sales representatives, and will result in either a successful sale (closed-won) or a failure (closed-lost).

1.2 Prioritization of leads

According to the company named SiriusDecisions, it is typical for only 6% of MQLs to convert to closed-won status. A major expense for sales teams is the time wasted on dealing with a large volume of low quality MQLs that will not become sales-qualified. In many cases, there will be more leads than can be targeted by the current sales team. Instead of hiring more teleprospectors, or arbitrarily choosing a subset of leads to pursue, sales teams should prioritize their efforts towards those leads that are most likely to reach the next stage. A predictive model can be employed for this prioritization. It can predict the probability of conversion, the probability of becoming closed-won, and/or the expected revenue from a given lead. The last of these allows a sales team to estimate the amount of sales and marketing budget that should be allocated to deal with particular leads.

The most expensive parts of the funnel are the sales qualification process and the actual sales process, that is sales representatives pursuing opportunities, since these stages require the most human work either by teleprospectors or sales representatives. Therefore, a predictive model can add the most value for these two steps of the funnel. Although this paper focuses on predicting lead conversion, the methods proposed are also directly applicable to ranking opportunities at earlier stages of the funnel.

Other reports of data mining techniques applied to sales and marketing include [2] and [1], which includes a chapter on identifying prospects using a CRM. Other analyses of using predictive techniques to gain insights into consumer behavior and to improve marketing operations are given in [11] and [3].

1.3 Conventional lead scoring

Lead scoring is not new. Many companies currently use a manual lead scoring system. Such methods are generally used by the marketing team to identify MQLs. Marketing automation software facilitates the creation of such scoring systems. Although the potential benefit of marketing automation has been recognized for at least 25 years [9], according to SiriusDecisions only 40% of sales teams with marketing automation think that their current marketing automation adds value. These systems still result in low quality MQLs being handed off to sales teams, making the sales qualification process expensive and time consuming. In this section we discuss these conventional scoring methods.

With a manual lead scoring system, scores are hand-tuned by experienced members of the marketing or sales team. These systems typically use a “scorecard” in which the presence or absence of certain positive or negative customer attributes or behaviors are assigned fixed positive or negative values. These individual values are then summed to determine a final score for the lead. For example, Table 1 shows some potential values that might be assigned for different behaviors and attributes.

Table 1: An example of a conventional lead scorecard. Values are traditionally hand-selected.

Behavior	Value
Filled out a contact form	+ 10
Attended webinar	+5
Visited webpage or blog	+1
Visited careers page	-10

Attribute	Value
Job title is “VP of sales”	+20
Company is located in Northeast USA	+5
Lead has a consumer email address (Gmail etc.)	-5
Job title is “student”	-10

One issue with conventional lead scores is that they fail to capture nonlinear effects. For example, if a user visits many webinars, they will receive a high lead score, since they accumulate 5 points for each webinar. However, there may be diminishing returns for each webinar visit. The highest quality leads may visit, say, between two and four webinars; attending additional webinars past this may not indicate a significant probability of making a purchase. It may even be the case that visiting many webinars is a negative signal. For example, it could indicate the behavior of a student, or even a competitor, who is researching the marketing functions of the company. In addition, complex interactions of features cannot be represented by scorecard models.

Another issue with conventional lead scoring is that the hand-selection of values is error-prone. In particular, hand-selection is vulnerable to bias from potentially mistaken business logic. This bias is also a problem for predictive methods, and is discussed further in Section 1.4.

A third disadvantage is that traditional lead scores are unbounded positive or negative values. They do not intuitively map to the probability of lead conversion or opportunity close. Many machine learning methods are probabilistic and therefore can give valid probability scores [13].

A fourth issue is that scorecard systems are often heavily reliant on behavioral data. While such data can be a good indicator of lead interest in the product, it prevents discovering the high quality leads early; they will only be found after enough time has passed for the lead to have taken specific actions. To avoid reliance on behavioral data, one could try to gather additional static features about the customer, but each additional feature adds complexity for hand-selecting an appropriate value.

1.4 Goals for automated lead scoring

The criteria for lead qualification vary greatly by seller. Determining that a lead is an MQL is usually based on simple behavioral and demographic rules. The demographic rules depend on the product or service being sold, and the behavioral rules depend on lead interaction with marketing materials specific to the company. As discussed above, identifying MQLs is an error-prone process, and the volume of MQLs is often greater than can be handled by the sales team. Even if there is not a great volume of leads, teleprospecting low-quality MQLs results in wasted time, and causes tension between the sales and marketing teams. This tension is the subject of research such as [8].

Most companies identify MQLs based on fixed criteria, usually not more sophisticated than a hand-tuned scorecard. Training a machine learning model could simply learn to reproduce these simple linear criteria, and therefore maintain the bias that is present in the existing, hand-tuned model. For example, if a company has focused its sales efforts on Florida, a machine learning model may learn that a prospect being located outside Florida is a negative signal, which may in fact not be the case. We describe below how our design reduces the effect of selection bias [12].

On the other hand, typically all SQLs are pursued by sales representatives. Therefore, there is little selection bias in the later stages of the funnel. This is a major reason why predictive models should be trained with information from later stages of the funnel. Another reason is that the ultimate goal of the sales funnel is to close a successful sale, even if the problem at hand is simply to find leads that are more likely to be qualified by sales.

In the design of the methods described below, we address several major goals:

1. A model should be probabilistic and have a meaningful interpretation, such as expected revenue or probability of successful close.

2. A model should not simply relearn an existing conventional lead classification model.

3. A model should be consistent with a separate opportunity won/lost classification model. That is, it should assign higher scores to leads corresponding to closed-won opportunities than to leads which convert but are not successfully closed.

4. The model should be able to find quality leads quickly, without relying too heavily on activity data.

The design of the models accomplishes goals 1, 2 and 3, while goal 4 is achieved by having good static (non-behavioral) features.

2. DATA FOR EXPERIMENTS

The data in our experiments is provided by Fliptop. It consists of sales and marketing data extracted from Salesforce and Marketo systems, to which Fliptop has appended additional proprietary features. As with conventional lead scoring, the type of features present are of broadly two kinds: static (or demographic) features and behavioral (also called activity) features.

The static features are information about either the individual contact or the company for which the individual works. Fliptop obtains some of these features directly from fields in Salesforce, and uses individual, company, and domain names from Salesforce to append additional features. These features include company level information such as industry codes, number of employees, market value, income, and company location. They also include company hiring features, such as the number of job openings in marketing, sales, business, and other departments. Fliptop appends binary features indicating which technologies are employed by the company. Such features include whether a company uses Java, marketing automation software, HTML5, etc. Finally, the contact's job title is appended as a categorical feature. These static features represent the fit of the individual and the product being sold. The majority of static features are binary or categorical values, and the remainder are numerical features.

Behavioral features represent actions taken by an individual, and capture interest in the seller by the potential customer during a specific period of time. These features are all numerical counts representing the number of times a user has performed a specific action that is tracked by marketing automation software. Examples of actions include visiting a product website, opening a marketing email, attending a webinar or trade-show event, and filling out a particular form, such as a product demo request form or an unsubscribe form.

The remainder of this section describes the data available for two sellers called Company A and Company B. This data consists of lead data pulled from CRM and marketing automation software, to which Fliptop then appended additional features. For additional information on data preprocessing, see Section 3.2.

Company A is a SaaS (software as a service) business with around 200 employees and annual revenue around $20 million. The training set for Company A consists of 5925 unconverted leads, 1320 leads that became closed-lost opportunities, and 1469 leads that became closed-won opportunities. For this company, we have 243 static features about leads and their employers, along with 350 behavioral features. The median closed price of a sale for Company A is $99, while the mean closed price is $9930 The mean is 100 times the median because the pricing varies greatly based on the product type and the number of software licenses sold. The variability in revenue makes identifying the best prospects for Company A especially challenging.

Company B is a software business with over 500 employees and annual revenue around $100 million. Its training set consists of 25904 unconverted leads, 956 leads that became closed-lost opportunities, and 1097 leads that became closed-won opportunities. For this company, we have 242 static features about prospects, along with 20 behavioral features. The median closed price of a sale is $29618, and the mean closed price is $46118.

3. DIRECT QUALIFICATION MODEL

The DQM (direct qualification model) 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 first describe the motivation for this model, then give details on how to construct and label a training set for it, and then describe the predictive algorithm.

3.1 Motivation

Predicting whether a lead will convert is a binary classification problem, and would seem to require only training a binary classifier. There are several reasons why this can be undesirable for lead qualification. The first reason is that it runs the risk of merely learning to reproduce the conventional lead scoring model that the company uses. Since traditional lead scoring models are typically scorecards with linear weights, machine learning models can predict lead conversion with high accuracy. However, this does not provide additional benefit to a sales team.

Another disadvantage of a two-class solution is that a lead that makes it further through the sales funnel is of higher quality than one that does not. Therefore, we would like scores to incorporate information about the chance of a lead to end up as a successful sale. If a lead conversion score incorporates closed-won probability information, it is more likely that the score will be consistent with a separate predictive model that ranks sales opportunities. That is, if lead A has a higher score than lead B, and both leads convert to opportunities A and B, we would like opportunity A to also have a higher score than opportunity B, according to an opportunity scoring model.

We address these potential disadvantages by classifying leads into three classes:

- NoCON: Leads that never convert.

- LOST: Leads that convert to opportunities that are ultimately lost.

- WON: Leads that convert to opportunities that successfully close (closed-won).

3.2 Training set

For the classes LOST and WON, we include leads that have closed within the last year, so that the model is up-to-date. The numbers given in Section 2 are after performing all the filtering described

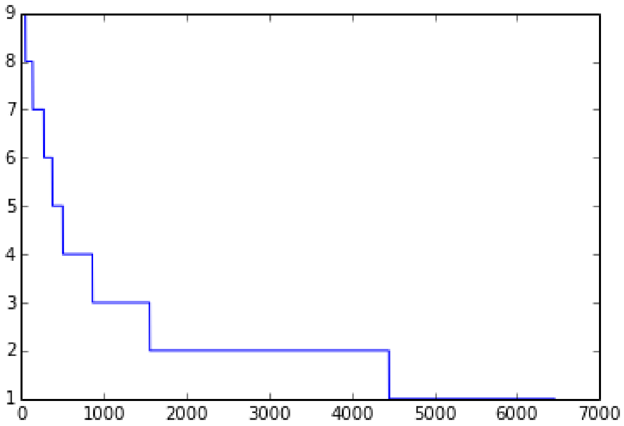

Figure 2: Leads are sorted by number of activities. The horizontal axis value is position in the sort, while the vertical axis value is the corresponding number of activities for that lead.

here. For the class NoCON, we use all leads that have not yet converted. While this class may contain a small number of leads that will eventually convert, that does not greatly affect the performance of our method. Another option would be to treat the unconverted leads as unlabeled, and use a positive-only learning method [4].

For behavioral features, we ensure that the only the most recent year of values is included; for most leads there is much less data than this. To avoid leakers [7], we only include activities that occur before conversion, and we remove activities that indicate actions taken by the marketing team, including administrative or data management actions, rather than by the actual prospective customer.

For company A, the great majority of unconverted leads have fewer than two activities, and similar features in general, meaning that a model could achieve high accuracy by simply identifying this great majority of unconverted leads. In order to investigate how our methods work well for companies with more variety in the class NoCON, we include all the leads with more than one activity, and a number L_1 of leads with fewer than two activities, such that L_1 is roughly equal to the number of leads with exactly two activities. Although this changes the distribution of leads, and therefore also changes the calibration of probabilities, this filtering of the training set is not unlike the process of clearing unpromising leads out of a leads database. Some companies will be more aggressive with deleting leads, so our method must work with different procedures.

For both Company A and Company B, we use 75% of the data for training and 25% of the data for testing. The training and test split was determined based on the time each lead was added to the lead management system. Leads in the test set were added after those in the training set, to approximate the real-world scenario where training the model occurs before lead scoring.

3.3 Training and prediction

We use a three-class gradient boosted tree classifier [5, 6]. The experiments in this paper use the implementation from scikit-learn [10] with the default parameters, and with so-called deviance loss in order for predictions to be probabilities.

After training the classifier on the training set, we use it to perform prediction on a separate test set. For each lead x to be scored in the testing set, the classifier gives three probabilities that sum to one: $p_1(x) = P(l = \texttt{NoCON}|x)$, $p_2(x) = P(l = \texttt{LOST}|x)$, and $p_3(x) = P(l = \texttt{WON}|x)$, where l is a label value conditional on x.

Table 2: AUC values for the DQM method.

Company	Features	AUC$_1$	AUC$_2$
A	All	0.992	0.960
A	Static only	0.988	0.940
A	Behavioral only	0.927	0.867
B	All	0.956	0.969
B	Static only	0.928	0.964
B	Behavioral only	0.906	0.922

Table 3: AUC values for the FFM method.

Company	Stage	AUC
A	Lead conversion	0.991
A	Closed-won	0.788
B	Lead conversion	0.952
B	Closed-won	0.912

There are several ways to map the three probabilities into a lead score $s(x)$. We consider linear combinations of p_2 and p_3:

$$s(x) = \alpha p_2(x) + \beta p_3(x).$$

After a linear combination is chosen, leads are sorted based on their score. As linear combinations we consider $(\alpha, \beta) = (0, 1)$, and $(\alpha, \beta) = (1, 1)$. These correspond respectively to maximizing closed-won probability and to maximizing lead conversion probability.

Although we only consider these two weightings, other weightings are possible. Alternative weightings may be desirable as a tradeoff between maximizing conversion of existing leads, which the marketing team is motivated to do, and maximizing closing of opportunities, which the sales team is motivated to do. The weighting can be tuned to demonstrate a sufficient benefit to both teams, which is important because companies who purchase predictive lead scoring solutions often split the cost between the marketing and sales budgets.

4. FULL FUNNEL MODELING

FFM stands for “full funnel modeling.” As a prospect advances in a sales funnel, he or she moves through several stages; see Figure 1. The FFM method uses a separate probabilistic classifier for each stage of the funnel, in whatever way the funnel is defined.

For companies A and B, the transitions we are most interested in are from lead to SQL (conversion beyond MQL) and from SQL to won (successful sale). We represent these transitions using two models:

$$P(\text{SQL}|x, \text{lead})$$
$$P(\text{won}|x, \text{SQL}).$$

Note that $P(\text{won}|x, \text{SQL}) = P(\text{won}|x, \text{lead}, \text{SQL})$ because SQL being true logically implies “lead" being true. Additionally, we include a third model for the final stage of the funnel, namely the size of the closed deal:

$$E(\text{revenue}|x, \text{won}).$$

In these expressions, x denotes the feature values describing a given potential customer.

The probability that a lead with characteristics x will become a successful sale is

$$P(\text{won}|x, \text{lead}) = P(\text{SQL}|x, \text{lead}) \cdot P(\text{won}|x, \text{SQL}).$$

The expected revenue from the lead is

$$E(\text{revenue}|x, \text{lead}) = P(\text{won}|x, \text{lead}) \cdot E(\text{revenue}|x, \text{won}).$$

Knowing the expected revenue from a prospect x at the stage when x is only a lead allows a sales organization to estimate better how much budget should be invested in pursuing this individual prospect.

A full funnel model can also make predictions involving prospects currently at the SQL stage. For example, the expected revenue from customer x given that x has reached this stage is

$$E(\text{revenue}|x, \text{SQL}) = P(\text{won}|x, \text{SQL}) \cdot E(\text{revenue}|x, \text{won}).$$

Separating the conversion classifier and the closed-won classifier results in another advantage of FFM. It is often the case that data about leads data and data about sales opportunities are stored in separate databases. In some cases, missing fields make it difficult to link up a lead with its corresponding opportunity, and vice versa. In such a case, an FFM can be learnt, while a DQM cannot, as we do not know whether to label converted leads as class `WON` or class `LOST`.

Filtering and preprocessing the features that describe prospects are done in the same way as described above in Section 3.2, but the training sets and labels differ. In general, FFM requires the construction of a separate training set for each transition that is modeled. Here, we have a training set of leads for modeling the probability $P(\text{SQL}|x, \text{lead})$ and a training set of opportunities for modeling $P(\text{won}|x, \text{SQL})$ and a training set of closed-won customers for modeling $E(\text{revenue}|x, \text{won})$.

We use the same classifier learning algorithm and parameters as in the DQM model, but for binary instead of three-class classification. For regression, we also use gradient boosted trees.

For FFM, we can compute the score $s(x)$ of a lead as either $s(x) = P(\text{won}|x, \text{lead})$ or $s(x) = E(\text{revenue}|x, \text{lead})$. The former definition is analogous to setting $(\alpha, \beta) = (0, 1)$ for DQM. Our experiments only consider scoring based on expected revenue of leads.

5. RESULTS ON REAL DATA

This section describes empirical results obtained from retrospective analysis of historical data. The next section describes results from actual use in practice of the DQM.

The historical data used for this section is described above. Experiments for DQM report two scalar evaluation metrics: AUC$_1$, the area under the ROC curve (AUC) for classification of non-converted versus converted leads, that is, class `NoCON` versus class [`WON` or `LOST`], and AUC$_2$, the AUC for the classification of leads that become closed-won opportunities versus those that do not, that is, class [`NoCON` or `LOST`] versus class `WON`. These correspond to ranking the leads with $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (0, 1)$, respectively. For FFM we report AUC for the two separate classifiers which predict conversion and closed-won. Note that predicting conversion is the same binary classification task with the DQN and FFM approaches, so AUC$_1$ and AUC for FFM conversion are in principle the same. Observed differences are due to randomness.

As another evaluation of score quality, we plot lift curves for each of the experiments that show the ratio of converted or closed-won leads as we increase the proportion of selected leads. We also include lift curves that show the expected revenue as we increase the proportion of selected leads.

5.1 AUC measures

Applying the DQM to Company A data results in the AUC metrics given in Table 2. In order to see how the different types of features contribute to the model, we give AUC metrics for a model built with all the features, one built with only behavioral features, and one built with only demographic ("static") features.

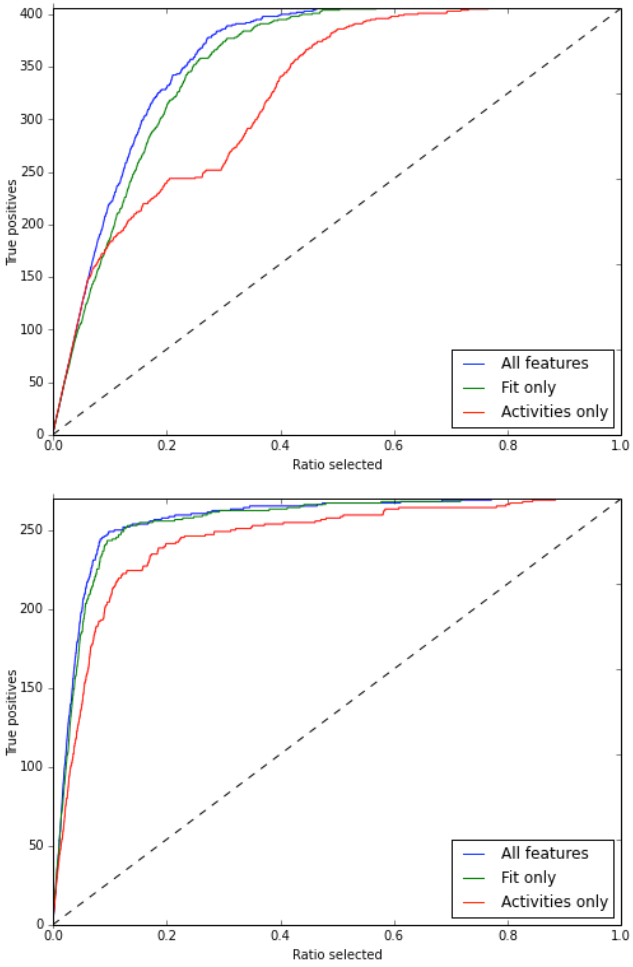

Figure 3: Closed-won lift curves for DQM with $(\alpha, \beta) = (0, 1)$. Top: Company A. Bottom: Company B.

AUC scores for the FFM method are given in Table 3. We show the AUC measures for the two classifiers: for predicting lead to SQL conversion, and predicting SQL to closed-won. To keep the paper shorter and more readable, we do not repeat the comparison of static versus behavioral features for FFM, and all FFM experiments use all behavioral and static features.

The AUC$_1$ scores are high. This is likely because the model can easily learn the existing business rules, that is the linear scorecard for qualifying leads. The way the DQM can add value over existing methods is by using further criteria to prioritize leads, as examined in lift curves for revenue and win rate shown below. A general reason why we are able to achieve high AUCs is that the training data includes all leads tracked in the CRM. Many of these are early stage leads, which are often obviously unlikely to convert.

5.2 Lift curves

To visualize the performance of DQM and FFM, we use lift curves. To understand these, note that the criterion for ordering leads on the horizontal axis is in general different from the quantity measured on the vertical axis. In particular, the DQM orders leads based on scores $s(x)$ corresponding to predicted probability

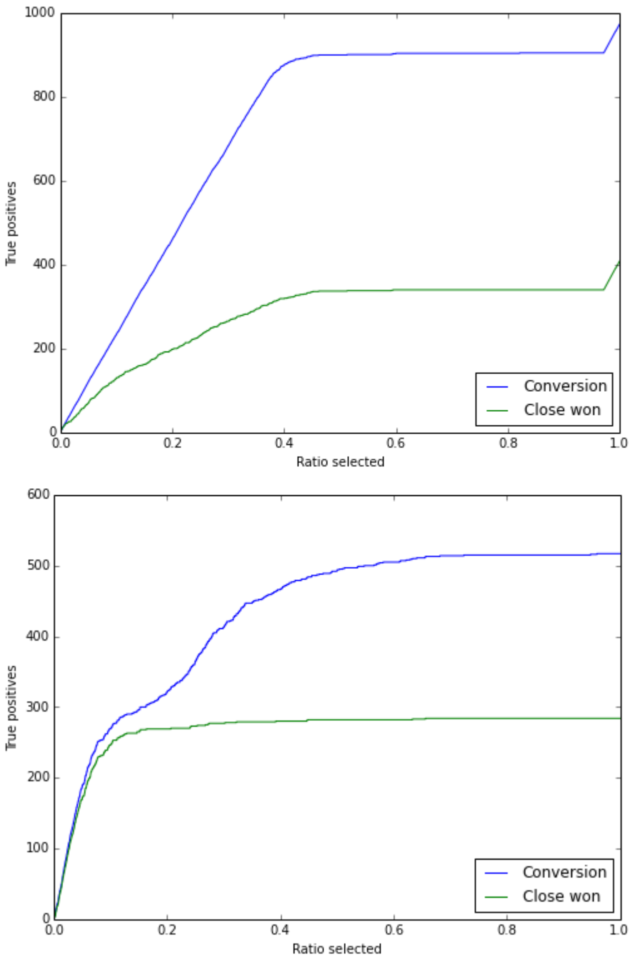

Figure 4: Conversion and closed-won lift curves for FFM. Top: Company A. Bottom: Company B.

of closed-won, using $(\alpha, \beta) = (0, 1)$. With this same ordering, we compute separate curves that track the number of successful sales and the sales revenue. Similarly, experiments with FFM rank leads based on expected revenue, but with this same ordering we again plot lift curves corresponding to number of conversions, successful sales, and the sales revenue. We use only one ordering for lift curves because this most closely matches the teleprospecting scenario, in which teleprospectors use a single ranking when contacting leads.

5.3 DQM and FFM comparison

Figure 3 shows closed-won lift curves for leads prioritized using $(\alpha, \beta) = (0, 1)$. It compares models obtained using all features, using just behavioral features, and using just static features. For both company A and B, we see that using all features performs best, while using behavioral features alone performs worst.

Figure 4 shows conversion and closed-won lift curves when we prioritize leads according to their expected revenue. For company A, the lift is significantly less in the 50% selected to 95% selected range, than it is in the 95% to 100% selected range. Figure 6 shows that the sales in this later range generate low revenue. It is often the case that bigger contracts have a lower chance of successful closing, but still a higher expected revenue overall.

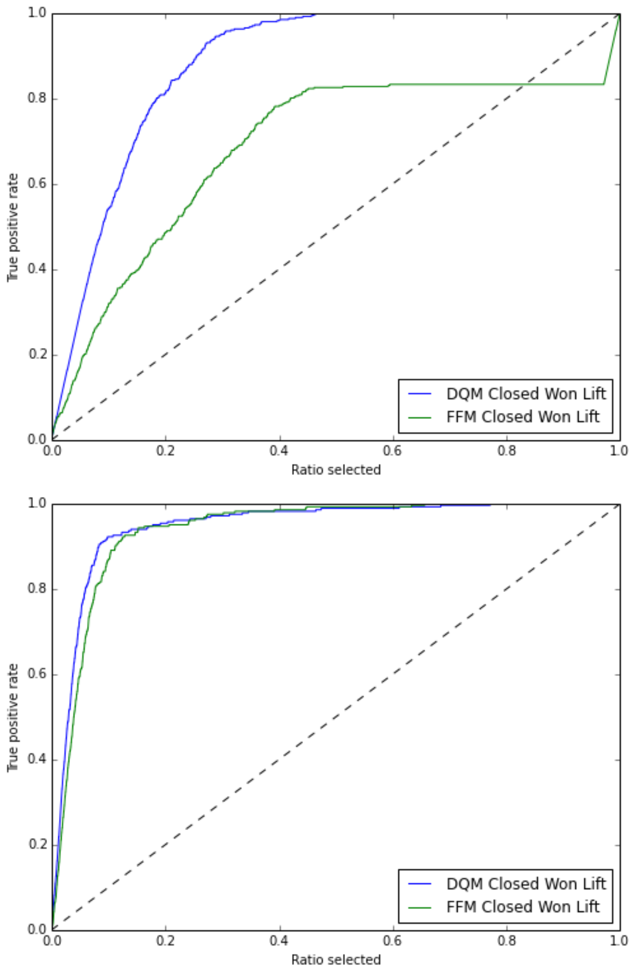

Figure 5: Closed-won lift curves, FFM versus DQM. Top: Company A. Bottom: Company B.

Figure 5 contrasts the closed-won lift curves for FFM and for DQM with $(\alpha, \beta) = (0, 1)$, both trained using all behavioral and static features. The ranking of leads for DQM is based on expected closed-won probability, while the ranking for FFM is based on expected revenue, so the closed-won curves are better for DQM. This is because the win probability for higher revenue deals tends to be lower, but the expected revenue is still higher for these deals.

Figure 6 compares revenue for the same models. For company A, DQM performs poorly at achieving lift in revenue. This is because the model focuses on closing the less risky, smaller magnitude sales. In general, the DQM method is less appropriate if there is high variance in the sales price. Alternatively, separate DQM models could be built for separate products or price ranges.

In Figure 5, the region to the very right of the FFM curve for company A (the straight line region) indicates that this method gives lowest priority to leads that with high confidence result in a low-revenue win. The DQM method achieves high initial closed-won lift for company A. However, the corresponding revenue curve in Figure 6 shows low initial lift, because DQM prioritizes low-revenue deals for company A. These observations suggests that it is easier to predict confidently low-revenue closes for company A.

As a final comparison, suppose that the sales teams of company A and B only have enough resources to contact 20% of all leads.

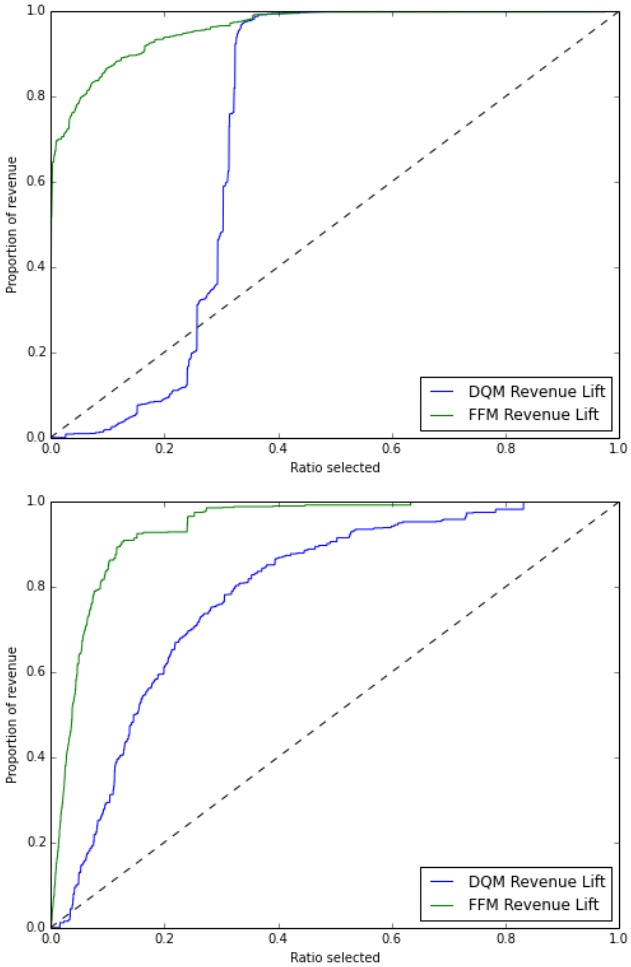

Figure 6: Revenue lift curves, FFM versus DQM. Top: Company A. Bottom: Company B.

Table 4 shows the conversion rates, closed-won rates, and revenue if the companies prioritize leads randomly, using DQM, or using FFM.

6. RESULTS IN PRODUCTION

Fliptop has deployed DQMs for several companies, and the initial results have shown definite benefits for both marketing and sales. Here we describe results for two sellers, called company C and company D.

6.1 Company C

The experiments with historical data show that we can best increase revenue using a full funnel model. However, the main goal of company C was to improve its marketing team's conversion rate, so a DQM was a good solution for it. Instead of giving explicit ranks, the DQM assigned four grades (A, B, C, D) by splitting the leads into deciles based on their ranking. These scores were then made visible in the sales team's CRM system.

The teleprospectors focused on qualifying A and B leads, and during three months (October 2014 to December 2014), Fliptop collected statistics on how the DQM improved the lead conversion rate and time needed to qualify leads. The conversion rate increased from 8% to 17% during these three months. In addition, the company found another unexpected benefit: the average time needed to qualify a lead went from 20 days to 7 days. This last statistic suggests that a three class model that takes into account the class `WON` produces benefits by accelerating the qualification process as well as by closing more sales.

6.2 Company D

This company provides a software tool to a variety of different types of customers. Fliptop built multiple DQM models for company D, with each model corresponding to a different customer type, or "vertical." The company and Fliptop worked together to collect statistics on the performance of DQM for the Dental and Lifestyle verticals. For two months (November and December 2014), they split the sales team into two groups: a control group that did not see Fliptop grades, and a test group that did see A, B, C, and D grades. The company and Fliptop took care that the two separate teams had similar historical conversion rates and rates at which they succeeded in scheduling demos for leads.

The team that used Fliptop for the dental vertical had a 31% higher conversion rate, and the lifestyle vertical had a 37% higher conversion rate. Additionally, the average number of times the sales team called a lead before conversion was 35% worse in the control group (35 calls versus 26 telephone calls) and the average number of demos scheduled per call placed was 28% better in the test group (3.6% versus 2.8%). These metrics indicate that the high quality leads uncovered by DQM are superior even in metrics that we are not optimizing over. The reduction in calls placed and in days to qualify a lead represent a significant decrease in workload for a sales team. Therefore the DQM has an even greater benefit impact, and potentially a positive impact in revenue, than the metrics in Section 5 might indicate.

7. CONCLUSION

This paper introduces two methods for modeling prospective customers moving through a sales funnel, called DQM and FFM. We examine how these two models can be used to perform predictive lead scoring. In order to provide benefit to a sales team, we design these models in such a way that they do not simply learn to duplicate a company's existing lead qualification rules, which can be error-prone and often do not take into account enough features. Instead we focus on predicting events further along in the sales process, in particular the likelihood of a successful close and expected revenue. Experiments show that applying these models to actual company data achieves high accuracy, both for classifying lead conversion and for predicting an ultimately successful future sale.

We also demonstrate that the model is predictive whether or not a lead has activity data, which means that high quality leads can be identified even before they take actions that can be tracked by the marketing team. We compare the two methods directly and conclude that FFM is more advantageous if there is more variance in the average sales price, because it can prioritize based on expected sales price, or if lead and opportunity databases cannot be reliably linked.

Production results show that DQM is successful in deployment, and that it has additional benefits for a sales team, separate from the metrics we are directly optimizing over. These include decreased time needed to qualify leads, and decreased number of phone calls placed by sales representatives in order to qualify leads and move them forward through the sales funnel.

Table 4: Expected results if companies A and B can only contact 20% of leads.

Company	Method	Conversion rate	Close rate	Total revenue ($)
A	Random	39.6%	16.6%	1,545
A	DQM	92.2%	67.6%	346,793
A	FFM	93.7%	40.3%	3,556,043
B	Random	8.4%	4.6%	2,619,493
B	DQM	27.7%	22.1%	8,009,377
B	FFM	26.1%	21.9%	12,149,051

8. REFERENCES

[1] M. J. Berry and G. S. Linoff. *Data mining techniques for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[2] I. Bose and R. K. Mahapatra. Business data mining—a machine learning perspective. *Information & Management*, 39(3):211–225, 2001.

[3] G. Cui, M. L. Wong, and H.-K. Lui. Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. *Management Science*, 52(4):597–612, 2006.

[4] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220. ACM, 2008.

[5] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[6] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[7] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):15, 2012.

[8] P. Kotler, N. Rackham, and S. Krishnaswamy. Ending the war between sales and marketing. *Harvard Business Review*, 84(7/8):68, 2006.

[9] R. T. Moriarty and G. S. Swartz. *Automation to boost sales and marketing*. Harvard Business Review, Reprint Service, 1989.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] M. J. Shaw, C. Subramaniam, G. W. Tan, and M. E. Welge. Knowledge management and data mining for marketing. *Decision support systems*, 31(1):127–137, 2001.

[12] A. Smith and C. Elkan. Making generative classifiers robust to selection bias. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 657–666. ACM Press, 2007.

[13] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the 18th International Conference on Machine Learning*, pages 609–616. Morgan Kaufmann, June 2001.