CRM Sales Prediction Using Continuous Time-Evolving Classification

Mohamoud Ali, Yugyung Lee
School of Computing and Engineering
University of Missouri - Kansas City
Kansas City, Missouri
mali@mail.umkc.edu, leeyu@umkc.edu

Abstract

Customer Relationship Management (CRM) systems play an important role in helping companies identify and keep sales and service prospects. CRM service providers offer a range of tools and techniques that will help find, sell to and keep customers. To be effective, CRM users usually require extensive training. Predictive CRM using machine learning expands the capabilities of traditional CRM through the provision of predictive insights for CRM users by combining internal and external data. In this paper, we will explore a novel idea of computationally learning salesmanship, its patterns and success factors to drive industry intuitions for a more predictable road to a vehicle sale. The newly discovered patterns and insights are used to act as a virtual guide or trainer for the general CRM user population.

Introduction

Apart from human capital and financial assets, data are now considered one of the highly regarded assets for any successful company. More and more companies are tapping into their ample data and translating them into actionable insights for market success. In this paper, we will explore a novel idea of computationally learning salesmanship, its patterns and success factors, to drive industry intuitions for a more predictable road to a vehicle sale.

Current predictive analytics research in the sales domain revolves around products or consumers. Recommendation systems and machine learning models for predicting product cross-sell, up-sell and deep-sell opportunities are examples of product centric analytics. On the other hand, consumer segmentation based on prior- or value-based attributes for targeted marketing campaigns is an example of consumer-centric analytics. Predictive analytics based on salesmanship and its success patterns are equally important and need appropriate research attention. Success patterns of salespeople can be turned into a product that can help the average salesperson stay on the road to a sale with more efficient continuity in a consumer relationship. With a greater focus on salespeople's activities to drive insights about success factors, we believe we are addressing a predictive analytics gap in the sales domain.

Our research also shows that timeline data with complex and evolving customer relationships do not fit well with current learning algorithms. In a CRM system, it is important for a salesperson to know what customers to contact. It is even more important to know when to contact them. In this paper, we present challenges and approaches for time-evolving classification problems.

Relevance and Significance

Predictive CRM is a newly emerging technology. It augments traditional CRM, which helps users sell or interact with known customers, with the identification of to whom or what to sell at the right time. Microsoft included AI capabilities in their Dynamics 365 CRM with the integration of Mintigo's Predictive Sales Couch (Mintigo 2017). SalesForce announced an AI-enabled, predictive CRM feature entitled 'Einstein' to make customer interactions smarter and help their clients focus on what matters the most at any given time (Salesforce 2017; Mintigo 2017). (Villacampa 2015) compares classification methods to identify potential buyers of new vehicles based on their vehicle service history. The trend, both in business and academia, shows artificial intelligence will continue to play a major role in the next generation CRM systems. However, as evident in the work cited above, current AI trend in the CRM industry draws insight from customer data and/or product sales. Other authors (Soltani and Jafari 2016) and (Wu, Guo and Shi 2013) describe that knowledge about the customer drives product insights. The mechanism to analytically customize products, present product recommendations or improve the consumer experience is realized through knowledge discovery for, from or about the customer. To our knowledge, there is no research work in the CRM domain that studies pattern discovery based on sale activities. One key aspect of our research is to drive sales insight from CRM user activities. We use successful and unsuccessful sales data to discover users' activity patterns. Our focus is on salesmanship. While customer and product insight are key business intelligence, we argue that we can change user behavior by studying their activities in addition to providing insight that drives sales up. Another key aspect of our study treats the temporal properties of time-evolving attributes as crucial predictors. We will cover this later in the paper.

Automotive sales regions of influence and problem of dimensionality

A vehicle's sale is generally affected by its brand, the dealership and prospective customers. These three components broadly represent domain dimensionality that influence knowledge discovery in the automotive industry. Each of these overly generalized dimensions encompass hundreds of other dimensions that can further influence a sale either directly or indirectly. Some of these influencers are shown in Figure 1. At the dealership, a successful sale or the lack of it is influenced by the effectiveness of its salespeople, business development center (BDC), finance and insurance department (F&I) and how effective its customer relationship management (CRM) works with the dealership management system (DMS). Through domain expert input and statistical observations, we selected twenty-three attributes for our sales prediction. We used the same technique to remove irrelevant and erroneous data, to narrow down the area of focus. In the following sub-sections, we show that statistical observations and data visualization can identify outlier regions and relational boundaries. This can help in filtering out noisy data and thereby reduce the issue of dimensionality. In the future, we wish to explore automated cross feature instance selection that will rank or filter a subset of selected instances based on its predictive power. In this paper, we will show our approach of defining noisy and outlier data boundaries and identifying any feature's predictive value range.

Figure 1: Sales regions of influence

Automotive data usually come from various resources, either through real time feed or historical data imports. It is inevitable that incorrect or incomplete data will make their way to most companies' databases. Missing data are easily noticeable. It can be treated using established data pruning techniques such as imputation or amputation (Wohlrab and Fürnkranz 2009). Incorrect data, however, are harder to detect during the machine learning workflow, and thus, inconspicuously undermine the effectiveness of training a model. Crude techniques, including manual data cleansing process by domain experts, do not provide a robust solution. We will show some techniques to detect incorrect data.

Sales Appointment

Setting up an appointment with an online sales prospect is important for an automobile dealership. In order to find the optimal interval within which setting up the appointment will positively influence the sales process, we collected 200,000 instances of actual lead data with appointments. We want to show the correlation between how soon an appointment is scheduled and how soon a lead is converted or sold. We also want to show where that correlation is weak. We extracted a numeric feature that represents the difference between when the lead was received (that is when the customer indicated an interest to buy a vehicle online) and when the appointment was first scheduled. We also extracted another numeric feature for the shelf age of all leads. If the lead was sold, the value will be the difference between when the lead was received and when it was sold. If the lead was not sold, we used the current date as the upper bound date time. Since we were studying the data from the prior year, unsold leads would normally have a shelf age of more than 300 days. Using the arithmetic mean over the nominal value {0, 1} for lead status where 0 indicates not sold and 1 indicates sold, we have confirmed the data were well distributed. Figure 2 shows the plot of the two features for the first 30 days of appointments and the first 60 days of leads. The light green dots indicate leads that were converted or sold. The red dots indicate the leads that were not sold. We observed that scheduling appointments was somewhat influential in the first thirty days after a lead was received. This observation underscores the need to analyze the degree to which a particular feature is predictive enough. Some researchers (Dash, Manoranjan, and Liu 1997; Guyon, Isabelle, and Elisseeff 2013) studied the overall usefulness or predictive power of a feature. For example, an attribute ranking technique may be used during attribute selection. Using this observation, we can argue that the same feature can have different predictive rankings. As a result, it is not enough to assign weights to attributes on a dataset level. A given attribute may not maintain the same ranking throughout its value range. This highlights the need for a careful analysis of correlation boundaries.

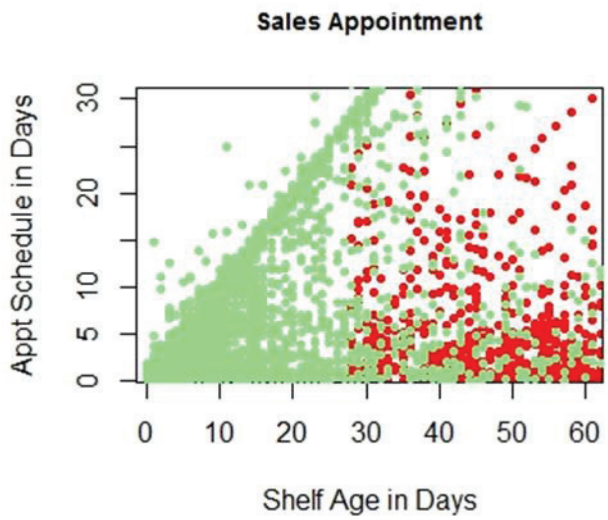


Figure 2: Historical data show appointment attribute loses predictive influence after 30 days

Customer Contact

At a time when consumers have more options, it is safe to assume that whoever contacts an online prospect first may have an edge of nurturing a sale relationship. The interval within which contacting the customer has the greatest impact, however, has not been established. In this section, we will explore the influence of an early contact to a sales prospect. The dataset we collected from the automotive CRM contains when a sales lead was received, when a salesperson attempted to contact the prospect, and when the actual contact, if any, was established. An attempt to contact the customer means a contact was made either through email, SMS or a phone call but an actual contact with the customer has not been established. We extracted a feature that represents the difference, in minutes, between when the lead was received and the first time a salesperson attempted to contact the prospect. We plotted this feature against the shelf age, and we found that the attempted contact time had more predictive power in the first 40 minutes of a lead lifecycle.

In this section, we showed that the traditional way of ranking attributes at a dataset level does not account for the range and variation among instances. We demonstrated that an attribute's rank can vary with the range of values in the dataset, underscoring the need to consider value range during feature selection or extraction.

Machine Learning Approach

Our goal was to use machine learning as a vehicle to expand the capabilities of an existing CRM system that serves over 4000 automotive dealers in the United States and Canada. While we had several use cases to choose from, sales prediction aligned well with the overall CRM function as a tool to drive sales results. A salesperson with insight of sales predictability is more likely to be efficient in how and when he interacts with his customers. The sales prediction use case also presented the opportunity to apply machine learning based on time-evolving user activities.

Sales prediction based on CRM Interactions

Successful salespeople follow a pattern. Learning algorithms can learn from successful salespeople's practices and patterns. We argue that interactions between the system users and the customers they are serving can contain good indicators for a successful sale. Today's diverse customer demographic allows for various ways to contact the customer including email, phone, text messaging and face-to-face meetings. Some of the interactions can be a system or a human agent acting on behalf of a salesperson. A BDC system agent may respond to customer enquiries for a vehicle sale during after-hours. We collected meta data for inbound or outbound messages, appointments, customer alerts, including price change alerts, and customer visits. Using various attribute selection tools including WEKA's attribute selection engine, we identified a total of 23 attributes for sales prediction.

Sales prediction based on time-evolving attributes

Machine learning algorithms heavily rely on the accuracy of selected attributes. Attributes or predictors contribute to knowledge gain or discovery from a hidden pattern. Selected or extracted feature attributes may contain unintuitive properties including geospatial and temporal properties that may contribute to knowledge discovery. The CRM data are usually a manifestation of a relationship that evolves overtime. This timeline dimension presents a challenge to incorporate the temporal property into an attribute's knowledge contribution. In this subsection, we highlight this challenge and limitation in current learning algorithms and present our approach of a continuous, time-evolving classification for better accuracy.

General classification learning algorithms learn from labeled data that include one or more variables and one target class. Each instance in the dataset represents an event. The learning algorithm discovers a pattern from the labeled data that allows for classifying unseen data similar in shape to the training data. Learning classifiers generally work to obtain a target class Y given X $P(Y \mid X)$ where $X = (x_1, x_2, ..., x_N)$ predictors or features and function P is the trust score, confidence score or probability. During our study, we observed CRM data did not intrinsically fit well with traditional classifiers. The time property of almost all features was as important as the features themselves. The time-evolving feature set is therefore $X = (x_1|t_1, x_2|t_2, ..., x_N|t_N)$ where t is the time property of x, or in other words, feature x given time t. Figure 3 illustrates the surface area of information gain when temporal properties are factored in during feature extraction. To further appreciate the information gain of time-evolving attributes, let us consider 'Appointment Set', which represents when a salesperson books an appointment with a potential customer. Treating this attribute as a nominal attribute 'IsAppointmentSet' with {0, 1} values contributes to the knowledge gain but not as much as if we consider timing. On the other hand, treating the attribute as discrete datetime will introduce noise during model training. A large number of leads, including new

Algorithm 1: Time-evolving feature extraction method

```
T0 ← conception time              ► (i.e. lead received)
if attribute is missing
    X ← ∞
else
    X ← time property of feature attribute
T ← current time ► (at induction or prediction time)
return Min (X, T) - T0
```

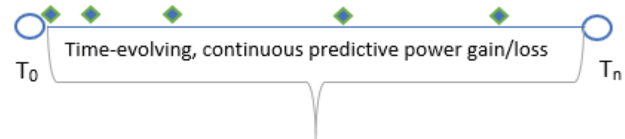


Figure 3: Illustration of attribute's temporal property for continuous classification represented by green dots

ones, may not have their appointment set and thereby causing missing value problem. We present an algorithm for time-evolving feature extraction. We globally define conception time $\mathbf{T_0}$, which represents the time any given instance or event is considered active. For our sales prediction, this is the timestamp when the customer made their initial enquiry to the dealership. For any given attribute, we identify its time property **X**. This is usually the attribute timestamp but it can also be a globally defined timestamp such as when a lead was sold. We then extract the desired feature by computing the difference between the event conception and current datetime or attribute timestamp, whichever is first. This method dramatically increased the accuracy of our predictive model. Empirical evaluation is presented in a later section.

Visualization Approach

Presenting sales prediction insights to existing CRM users was a key target in our study. However, certain users may interpret predictions at face value and avoid working on sales leads with low sales prediction scores. It is tempting to only favor sales leads with high prediction scores. We wanted to provide the user with predictive insights while avoiding attaching negative connotations to low scoring sales leads. To counter that possibility, we opted for a predictive badging system. We reiterate that one of our research goals was to use artificial intelligence to enhance an existing CRM system in helping keep the salespeople engaged with their customers by giving them more insight into how their activities are contributing towards a sale. We asked a group of experts in various departments, including UX design, product development and software engineering to weigh in and vote on badging options. The final badging system for our sales prediction is shown in Table 1. We show the user interface for the lead page with prediction UI in Figure 4. We used sample customer names to protect customer data. When an application user hovers over the prediction badge, we display additional recommendations to improve the score. Periodically, a software service agent sends each CRM user's activities and lead information to the sales prediction engine that runs in the Amazon cloud environment. Once the data are run through a trained model, the prediction engine sends the result back to the CRM system agent. The result contains the lead identifier,

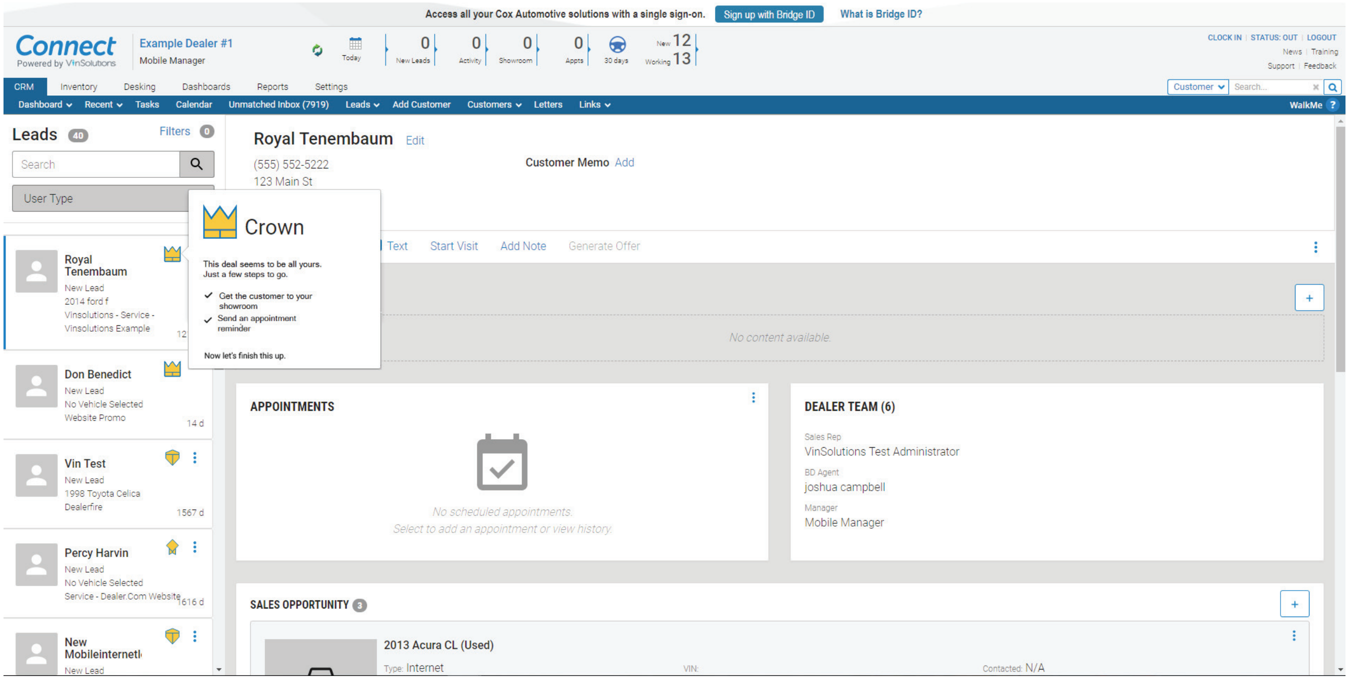


Figure 4: Lead page with predictive badges

predicted sales lead status (i.e. whether it was predicted to sell or not) and the confidence score of a predicted sale. The classification algorithm returns a confidence score for each class. For example, if the problem is a binary-class classification, we expect two independent confidence scores for either class based on the input data. The sales prediction model classifies the input data into a sold or active class and assigns a confidence score to each class. For our visualization, it suffices to get the confidence score of 'Sold' class and translate that to a predictive badge. The badges represent a sense of achievement due to the user's activities. A new sales lead will always have a very high prediction score. The trained sales prediction model recognizes that a sales lead at this stage has a good chance of success, not only because it is a fresh lead but also because the salesperson is automatically on track in the sales journey. Even though the confidence is normally very high for new leads, we chose to assign a neutral predictive badge. If a sales lead is not new and still maintain a high prediction score of 95% or more, we assign the crown badge to indicate the salesperson competitively pursued this lead and is likely to close on the deal. We allow dealers to configure the threshold, but it must stay within the range of 85-100%. Likewise, we assign the ribbon to any low scoring sales lead to encourage users to keep pursuing the lead until they make up their mind not do so. The shield badge represents both a medium score and the need to protect this deal opportunity as it can easily slip away. In the case of medium score, we wanted to provide more insight on whether the user is trending up or down based on his activities (or lack of activities). We take the average of the most recent three predictions of a given lead and compare it with the last obtained confidence score to determine downward or upward trend.

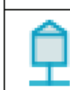
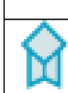
Predictive Badge	**Label**	**Role of Confidence Score**
	Neutral confidence score	Assigned to new leads regardless of their score.
	High confidence score	Score above 95%. Threshold configurable.
	Medium Confidence – Trending Up	- Score is medium. - Average score of the three prior predictions is greater than the new score.
	Medium confidence score	Configurable range with default range of 75% to 95%.
	Medium Confidence – Trending Down	- Score is medium. - Average score of the three prior predictions is less than the new score.
	Low Confidence score	Configurable value with default score of 75% or less.

Table 1: Predictive Badges for User Interface

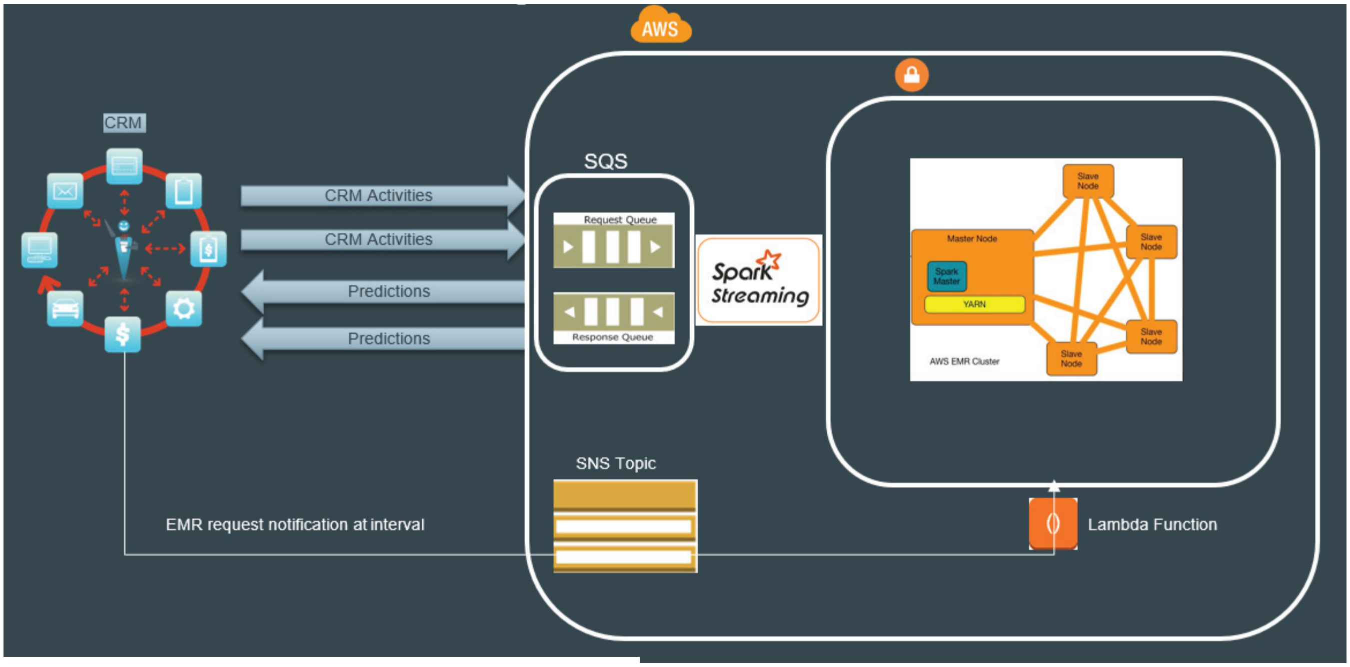


Figure 5: Sales prediction system architecture

Sales Prediction Workflow, Deployment and Support

Mindful of the highly transactional, largely on-premises CRM system and the potential CPU spike due to the compute-intensive sales prediction, we established key requirement drivers for the machine learning environment. We adopted a cloud-first strategy for managing cost, solution portability, out-of-the-box reliability, resiliency, security, and a disconnected yet highly integrated compute. Simplicity was another key requirement driver. We chose Apache SparkML for its support of various programming languages including Java, Scala and Python. Amazon's Elastic MapReduce (EMR) works with SparkML. Figure 5 shows the sales prediction workflow and system architecture. The workflow starts with scheduled jobs that runs on-premises within the CRM system. The jobs are hosted as a Windows-based service agent. One job queries the database, prepares the machine learning feature set and pushes it to an AWS Simple Queue Service (SQS). We chose SQS over AWS Kinesis for its simplicity and lower cost. We did not need to replay the same sales prediction request; something that may have been a better candidate for Kinesis. As the agent job sends data to SQS, it also sends a notification to AWS to request EMR resources. This is done via publishing a message through AWS Simple Notification Service (SNS). We configured a lambda function that gets triggered when the message is published. The lambda function will allocate EMR resources, which includes a Spark cluster with one master node and two core nodes. AWS S3 is used for logging. It also stores versioned JAR files and trained models. The seamless integration of AWS services allowed us to spin up an entire production environment in less than 5 minutes. The sales prediction takes less than 90 seconds at production scale of over 2.5 million predictions. The EMR cluster is configured to tear down itself after 10 minutes since the last detected streaming activity.

The product, as previously shown in Figure 4, is fully integrated with the Cox Automotive's Connect CRM. Technical support is, therefore, handled by the same CRM support team. Customer feedback and feature requests are received through the performance management team. We retrain underlying machine learning models every quarter or when new features are added. Production grade models are protected with strong AWS authorization access policies. Approved automated builds using terraforms can only update the production models. An extensive series of evaluations must be completed before a model is promoted to production. We discuss the evaluation process in the next section.

Evaluation

Given that the project was targeting CRM subscribers, we conducted four rounds of evaluation throughout the study. We started with a series of machine learning evaluations as we experimented with a varying number of features until we found a highly predictive feature subset, an algorithm that fit with our dataset, and a model that consistently performed well. We trained our machine learning model with 300,000

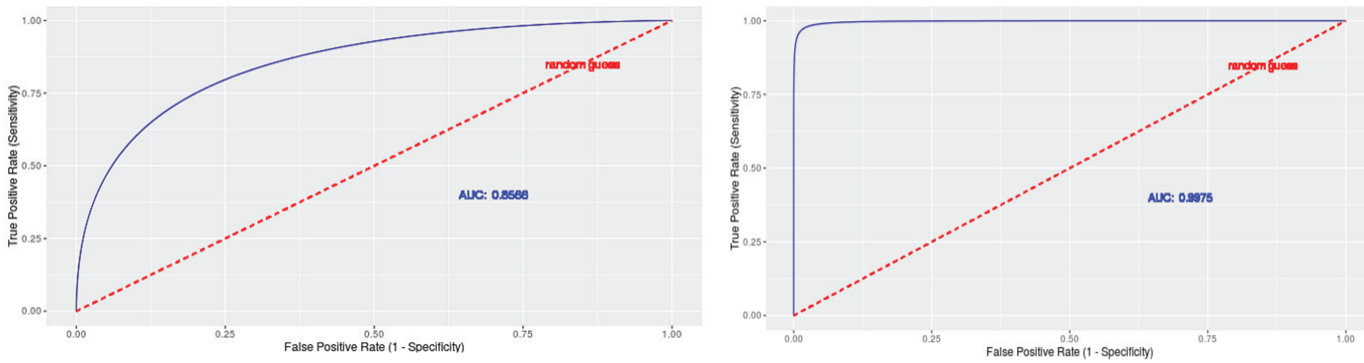


Figure 6: Area under ROC curve before and after continuous time-evolving classification

instances of actual sales leads from a highly utilized CRM system. The data were anonymized and changed to numeric representations before they were released to the research team. The training data were well distributed with approximately equal proportions of sold and active sales leads. The {Active, Sold} nominal value was the target variable or class in a dataset that contained 22 other variables. That gave us a training dataset of 6.9 million features. We experimented with Naïve Bayes, Logistic Regression and Random Forest. Logistic regression was performing very poorly and was dropped from the study. Naïve Bayes performed fairly well with 86% accuracy, but not as well as Random Forest with 99% accuracy. Naïve Bayes normally fits well with data that have mostly independent variables. Our dataset included correlated variables such as when the appointment was set and when it was confirmed. We ultimately chose Random Forest because it consistently performed better, converged faster and naturally responded to our massively parallel processing EMR jobs. The algorithm's way of creating iterations or forests made it a natural fit for parallel processing. Table 2 shows the evaluation metrics. Figure 6 shows the area under ROC curve before and after using continuous time-evolving classification approach. Our approach improved detection probability, also known as true positive rate, and reduced the false-alarm or false positive rate as shown in the right graph of Figure 6. The ROC analyzes detection strength as a function of misclassification cost. Any curve above the

Measurement	Classification	Continuous time-evolving classification
Accuracy	89%	99.6%
Precision	81%	98.7%
Recall	87%	99.1%
Area under ROC	85%	99.9%

Table 2: Sales Prediction Metrics

redline is better than a random guess and vice versa. The classification of a model performs better as its ROC curve moves further to the top and left above the red line, thereby creating wider area under ROC curve (AUC).

In the second round of our evaluation, we deployed our machine learning software, collected 20,000 instances from

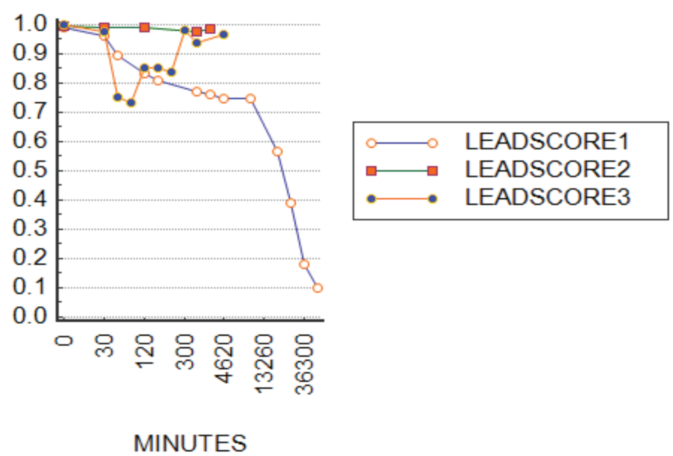


Figure 7: Sales prediction trend evaluation for three leads

sales leads sold two years ago, and run it through the trained model. The model correctly classified 98%, approximately matching the ROC measurement obtained in the prior evaluation. In the third round of evaluation, we have asked product managers and quality assurance analysts to track the sales prediction performance for three weeks. We iteratively retrained and evolved our predictive model based on feedback we received during the evaluation period. In Figure 7, we show three different sales leads we selected out of dozens of leads we tracked during evaluation and quality assurance testing. We show how the sales prediction confidence score (Y-axis) changed over time (in minutes) based on the interactions. The machine learning model usually assigns all new leads a high score. That is because our approach during feature selection and extraction put an emphasis on the timeline. Without interaction, the prediction score drops significantly after 2-3 days and may reduce to zero after 30 days. We do not show the confidence score to the salesperson to avoid cherry-picking. Rather, we translate the score to a predictive badge that subtly let the user know whether they are on track or not. We observed the use of predictive badges increased the lead page utilization by 15%. Finally, we piloted the product feature with 2,200 active users in 10 geographically disperse dealer locations. We plan to exhibit this product at the National Automobile Dealers Association (NADA) 2018 convention.

Conclusion

In this paper, we presented sales prediction for automotive CRM based on customer interactions. We discussed the merits of introducing a new machine learning approach for sales prediction that is salesperson centered as opposed to customer or product centered. We also studied the challenges and opportunities of time-evolving attributes in pattern discovery. Through a series of evaluations, we confirmed that not only our predictive model was accurate but also our visualization approach was intuitive and easily consumable by CRM users without additional training. This new predictive capability serves to keep the salespeople on the right track. The product learns from the entire user population of the CRM system and translates best practices and patterns into easily consumable knowledge.

Our future efforts focus on expanding this work to sales scenario prediction. We wish to train models that can identify and rank multiple sale scenarios and recommend salesperson's next best route on the road to a sale. We also plan to integrate this feature with the Business Process Management (BPM) product line of the CRM system. This will allow for business processes that automatically fire based on prediction results. One example is sending a notification to the sales manager when a lead with a high predictive score has not be contacted for a certain period. Other product lines including after-sale marketing, campaigns, and task management are being considered for integration with the sales prediction.

Acknowledgement

VinSolutions, a Cox Automotive brand, funded this project with a research grant (number 0049303) to UMKC Distributed Intelligent Computing Center. UMKC Distributed Intelligent Computing laboratory provided resources including servers and access to research materials.

References

"Mintigo Delivers Artificial Intelligence for Microsoft Dynamics 365." Journal of Robotics & Machine Learning (2017): 78.

Salesforce. "Salesforce Introduces Service Cloud Einstein, the World's #1 Intelligent Customer Service Platform." Journal of Engineering (2017): 1074.

Villacampa, Osiris. Feature selection and classification methods for decision making: a comparative analysis. Diss. Nova Southeastern University, 2015.

Dash, Manoranjan, and Huan Liu. "Feature selection for classification." Intelligent data analysis 1.1 (1997): 131-156.

Soltani, Zeynab, and Nima Jafari Navimipour. "Customer relationship management mechanisms: A systematic review of the state of the art literature and recommendations for future research." Computers in Human Behavior 61 (2016): 667-688.

Wu, Jiebing, Bin Guo, and Yongjiang Shi. "Customer knowledge management and IT-enabled business model innovation: A conceptual framework and a case study from China." European Management Journal 31.4 (2013): 359-372.

Wohlrab, Lars, and Johannes Fürnkranz. "A Comparison of Strategies for Handling Missing Values in Rule Learning." (2009).

Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." The Journal of Machine Learning Research 3 (2003): 1157-1182.

Husain, Ahraz. Understanding How Developers Work on Change Tasks Using Interaction History and Eye Gaze Data. Diss. Youngstown State University, 2016.

Molina, Luis Carlos, Lluís Belanche, and Àngela Nebot. "Feature selection algorithms: a survey and experimental evaluation." Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. IEEE, 2002.

Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." Computers & Electrical Engineering 40.1 (2014): 16-28.

Tang, Jiliang, Salem Alelyani, and Huan Liu. "Feature selection for classification: A review." Data Classification: Algorithms and Applications (2014): 37.

Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." Pattern Analysis and Machine Intelligence, IEEE Transactions on 27.8 (2005): 1226-1238.

Khalid, Hammad, et al. "What do mobile app users complain about?." Software, IEEE 32.3 (2015): 70-77.

Pellet, Jean-Philippe, and André Elisseeff. "Using Markov blankets for causal structure learning." The Journal of Machine Learning Research 9 (2008): 1295-1342.

Cao, Yi, et al. "Adaptive hidden Markov model with anomaly states for price manipulation detection." Neural Networks and Learning Systems, IEEE Transactions on 26.2 (2015): 318-330.

Wagstaff, Kiri L., and Victoria G. Laidler. "Making the most of missing values: Object clustering with partial data in astronomy." Astronomical Data Analysis Software and Systems XIV. Vol. 347. 2005.

THINK IT Services: Can Salesforce Einstein Catch the Next AI Wave? A Guest Commentary in TechTarget. Newstex, Chatham, 2016.