

IT@INTEL

Data Mining Using Machine Learning to Rediscover Intel's Customers

Intel IT developed a machine-learning system that doubled potential sales and increased engagement with our resellers by 3x in certain industries.

Eiran Bolless
Senior Data Scientist,
Advanced Analytics, Intel IT

Jeremie Dreyfuss
Senior Data Scientist,
Advanced Analytics, Intel IT

Amit Kalfus
Senior Developer,
Advanced Analytics, Intel IT

Shahar Shpigelman
Regional Sales Group Analytics,
Sales and Marketing

Shahar Weinstock
Data Scientist,
Advanced Analytics, Intel IT

Executive Overview

Data mining using machine learning enables businesses and organizations to discover fresh insights previously hidden within their data. Whether exploring oil reserves, improving the safety of automobiles, or mapping genomes, machine-learning algorithms are at the heart of these studies.

At Intel, we are quickly moving machine learning from an academic pursuit to a driver of innovation and competitive advantage for our business. To that end, Intel IT developed a machine-learning tool that helps Intel's sales and marketing organization identify which resellers will best connect with customers in specific vertical industries.

The machine-learning algorithm helps us learn more about our resellers by classifying them and then supplementing this information with another algorithm that mines resellers' website content. The machine-learning system can also integrate data from an Intel® product-recommendation system.

We conducted a proof of concept (PoC) that demonstrated that our tool was effective even when users invested little effort and that it worked across geographies and languages to identify relevant resellers.

Since the PoC, Intel sales and marketing teams have delivered communications to eight vertical industries across four geographies and in eight languages. When we tracked resellers in the engagement chain, we found that, in comparison with the rest of the sales pipeline, twice as many resellers advanced from leads to qualified leads. Their click-through rate for email newsletters is now three times higher, and they complete Intel training at a rate three times higher than the rest of the sales pipeline.

These results reinforce Intel IT's growing use of machine learning to turn big data into deep insights and to make recommendations in real time for our smart and connected world.

Contents

- 1 Executive Overview**
- 2 Background**
 - Business Challenge
 - Expanding Uses for Machine Learning in the Industry
- 4 Solution**
- 4 Solution Architecture**
 - Platform Training Approach
 - Data Considerations
 - Classification with Imbalanced Data
 - Classification Based on Little Data
 - Model Transfer between Languages
 - Initial Results from Our Proof of Concept
- 13 Results: Positive Business Impact**
- 14 Conclusion**

Acronyms

CRM	customer relationship management
HDFS*	Hadoop* distributed file system
IG	information gain
IoT	Internet of Things
Odds	odds ratio
PoC	proof of concept
QFS	query feature selection
SVM	support vector machine
tf-idf	term frequency-inverse document frequency
X²	chi-square

Background

Intel primarily uses an indirect sales model. Intel sells products to resellers, who then integrate them into other products, which may in turn be sold by yet another company as part of a broader solution or service. As technology rapidly expands into new industry segments, Intel sales and marketing teams strive to help resellers operate in those segments.

Business Challenge

Intel's sales and marketing organization strives to increase communications with resellers relating to new industry segments as they evolve and to encourage resellers to join webinars or conferences to gain a better understanding of Intel's offerings for these new markets. Because sales and marketing teams must focus their resources on those resellers with the highest probability of generating sales, sending the right message to the right reseller helps drive value for the sales pipeline.

Our sales and marketing organization's data-source ecosystem is complex. Many large resellers have multifaceted relationships with suppliers, with multiple salespeople serving disparate needs. Surveys, insights, or data from distributors, resellers, or customer relationship management (CRM) systems tell only part of the story. Public-facing information is often incomplete or inaccurate.

Helping Intel sales and marketing teams acquire a clear, up-to-date view of Intel resellers and their business direction provides our teams with the ability to build a connected sales pipeline. To provide transformative insights that could increase revenue opportunities, Intel's sales and marketing organization needed to better understand our resellers and their customers. We needed a machine-learning system to help our sales and marketing teams identify the best prospects within Intel's large pool of resellers.

Expanding Uses for Machine Learning in the Industry

The theory of machine learning is not new, but its potential has been largely unrealized due to the absence of the vast amounts of data needed to make machine learning useful. The recent explosion of big data, however, has made data mining using machine learning one of the most active areas of predictive analytics.

Machine learning is an outgrowth of artificial intelligence. It enables researchers, data scientists, engineers, and analysts to automate analytical model building by constructing algorithms that can learn from and make predictions based on data. Within Intel's business, machine learning can be used for modeling web-browsing behavior, identifying the best resellers to increase sales, providing just-in-time decisions from sensor-based manufacturing, and detecting financial fraud—to name just a few relevant applications.

Machine learning enables enterprises to discover patterns and trends from increasingly large datasets and also enables them to automate analysis that has traditionally been performed by humans. It provides higher confidence levels for strategic decisions and recommended actions. With data-defined models, enterprises can deliver differentiated and personalized new products and services, as well as lower the cost of existing offerings. Machine-learning initiatives should be considered not only as strategic initiatives but also for their possible effects on business strategies.

All of these factors mean it is possible to quickly and automatically produce models that can analyze more complex data and deliver faster, more accurate results—even on a very large scale. The result is high-value predictions that can guide better decisions and smart actions in real time, without human intervention.

One key to producing smart actions in real time is automated model building. Analytics thought leader Thomas H. Davenport wrote in *The Wall Street Journal* that with rapidly changing, growing volumes of data, “you need fast-moving modeling streams to keep up.” And you can do that with machine learning. He says, “Humans can typically create one or two good models a week; machine learning can create thousands of models a week.”¹

Unleashing the power of machine learning requires certain ingredients: access to large amounts of diverse data, optimized data platforms, tools, and the skills to build the platforms. In addition, machine learning requires a highly scalable, flexible infrastructure—including compute, memory, storage, and network—on which to develop, train, and deploy analytic models.

There are two parts of the machine-learning process: training and scoring (for more information, see the sidebar “What Is Machine Learning?”). In the training phase, model developers “teach” the computer by processing huge amounts of data with embedded clues. The computer creates a model that can be executed by end devices—like sensors, PCs, or smartphones—to obtain more data for training. The crucial aspect is that as the model is being trained, the results are scored, or evaluated, and fed back into the training process so that the system continues to learn and the model improves.

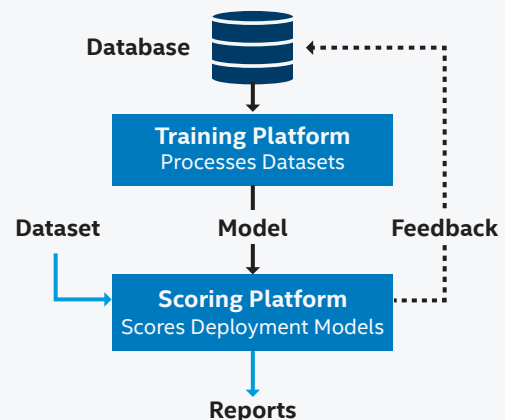
What Is Machine Learning?

Machine learning is defined as “extracting features from data in order to solve predictive problems.” Programs learn to recognize complex patterns automatically and then make intelligent decisions based on insight generated from learning.

For accuracy, models must be trained, tested, and scored to detect patterns using previous experience:

- Training builds a mathematical model based on a dataset. This includes four steps:
 1. Select the relevant dataset.
 2. Query the dataset.
 3. Manually label the query results.
 4. Train the classification model.
- Testing evaluates the quality of the model. Testing queries manually labeled data that was not used in training.
- Feedback uses labeled data from testing to add updated data to the training phase.
- Scoring uses the trained model to make predictions about new data. Several scoring models can be deployed, including classification, prediction, detection, and recognition.

In our use case, we used machine learning to classify companies, their web content, and language translation. Sales and marketing users label returned web pages as relevant or nonrelevant to their industry segment query (training). Then web pages are tested and scored, using data from resellers' websites. The output is a list of resellers best related to users' queries, such as “best resellers for smart buildings” or “best resellers for solutions using Intel® Xeon Phi™ processors.”



Machine learning creates analytical models using training and scoring phases to gain insights not possible with less advanced data-analytics methodologies.

¹ Source: www.sas.com/en_us/insights/analytics/machine-learning.html

Solution

Intel IT developed a tool named “Reseller Knowledge Base” to help Intel sales and marketing teams tap into Intel’s customer base and identify the resellers that offer the highest probability for sales. To offer a general-purpose knowledge base system, we created a multilayered approach composed of three parts (see Figure 1):

- **WebInsights.** This web-mining tool allows users to train a semantic model using a search query and subsequently label pages from any reseller or potential reseller’s website. By evaluating reseller website content, WebInsights tells us what resellers tell their customers (Phase 1 in Figure 1).
- **ResellerInsights.** This predictive system uses Intel’s CRM information and learns from the output of the WebInsights tool to find patterns in the data that can predict which resellers are most likely to respond to marketing campaigns. ResellerInsights also reveals reseller focus by evaluating the Intel training that the resellers take and the questionnaires they complete.
- **SMART Target.** This reverse recommendation system finds the resellers most likely to buy and sell Intel® products by unveiling resellers’ buying patterns (Phase 2 in Figure 1).

The advantage of this multilayered approach is that it allows us to integrate all the insight we have gained about Intel resellers.

Solution Architecture

Our machine-learning platform (see Figure 2) is designed to be highly scalable and to quickly process vast amounts of data. It combines database software with pretuned Intel® Xeon® processor E5-2600 v4 family-based servers at a cloud-provider host. Training and scoring of the datasets run in a single environment.

The platform is comprised of eight servers. Three servers are dedicated to search queries and navigation using Apache Solr*. Two servers manage web services for the Reseller Knowledge Base for both front- and back-end-related services (based on Python* and Java*). In addition, three servers run Cloudera* distribution for Apache Hadoop* to manage storage and compute.

The Reseller Knowledge Base platform is currently using 1.5 TB of data distributed on a Hadoop* distributed file system (HDFS*) with replication and high-availability features. This structure is designed to scale as data increases.

The software stack starts with the base Cloudera distribution for Apache Hadoop software, on top of which we run several applications: Apache Nutch* with a YARN* web-based user interface for the web crawling and scrapping, and Apache Solr for indexing and searching web-page text. Java and Python microservices run the algorithms and data manipulations. In addition, the platform uses various open-source libraries and solutions.

Reseller Knowledge Base Process

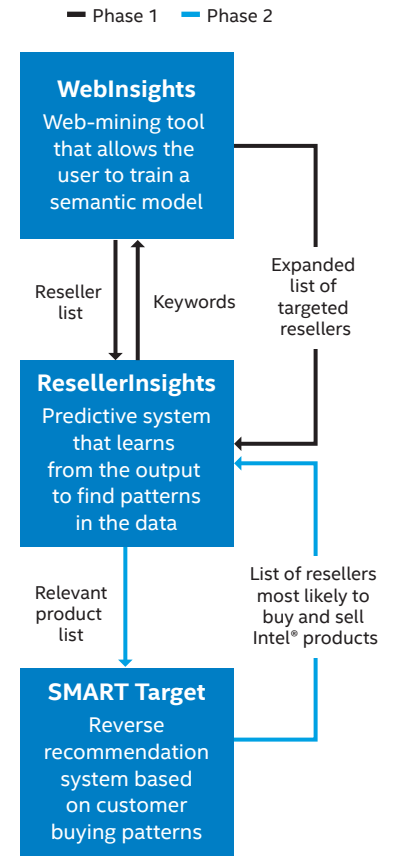


Figure 1. Intel’s Reseller Knowledge Base platform is a multilayered system composed of WebInsights, ResellerInsights, and SMART Target tools.

Machine-Learning Platform Built on Intel® Technology

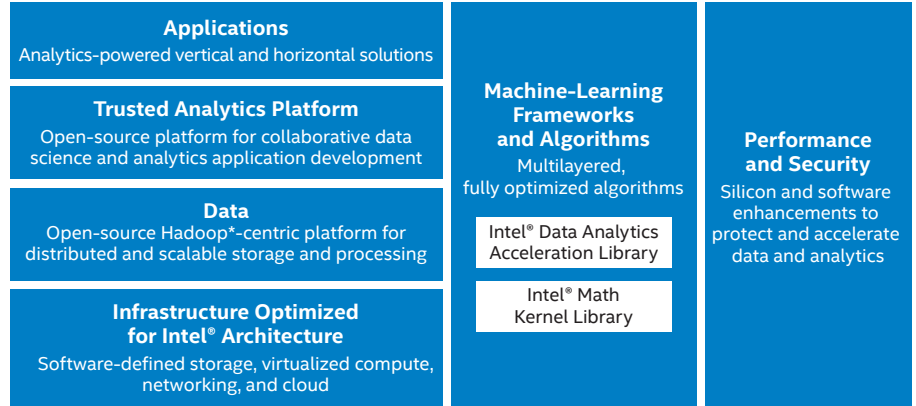


Figure 2. A fully optimized machine-learning environment is built on tightly integrated Intel® technologies for accelerated insight discovery at a lower cost of ownership.

Platform Training Approach

Upon launching the Reseller Knowledge Base platform, users enter keywords that describe target companies, geographies, and languages. Because WebInsights is based on website content, it identifies both Intel resellers and companies who are not affiliated with us.

Because users do not always have a clear picture of the companies with whom they want to work in new industry segments, the keywords they use do not always capture their needs. The system alerts users to these issues through a feedback file, which details the coverage of each query term. Often users select words that appear on few if any of our resellers' websites. This means either no company is involved in that particular aspect of the industry or, more likely, the way companies describe the segment is different from the way users thought of it. Users can refine the query, or they can proceed to reviewing pages from those companies whose websites the query captured.

To help users find alternative query words to define the targeted segment, WebInsights returns a list of the most similar keywords. Once users are satisfied with the query results, the system provides pages to review and label—either as positive (relevant) or negative (not relevant) with respect to the industry segment.

WebInsights uses these labeled pages to learn a classification model and return to users only relevant companies. However, users are generally willing to review only 20 to 40 pages. This means that to define the segment, we have to solve a binary text-classification problem with very little training data.



To help users find alternative query words to define the targeted segment, WebInsights returns a list of the most similar keywords.

The fact that individual users are interested in different segments of a given company forces the system to operate without a taxonomy of categories. This allows every user to find exactly what he or she is looking for without being limited by predetermined categories; however, it also prevents the system from learning segments in advance or pooling labels of pages from individual users.

Intel has sales teams and resellers around the world. Sales teams in Germany cannot afford to overlook companies in France or England due to language differences. On the other hand, we cannot expect German-speaking users to find French keywords to define their queries and then review a set of web pages to label them as relevant or not. This is why the ability to review pages in one language and label pages regardless of language is important.

When users are satisfied with WebInsights results, the system feeds those results to ResellerInsights. Because some Intel resellers do not have a website, defining a relevant understanding of companies using WebInsights alone would not work. Furthermore, integrating firm demographics and reseller interactions can provide additional insight. ResellerInsights combines the meaningfulness of the web-mining approach with the strengths of firm demographics and reseller information to help identify and rank relevant resellers according to their value for potential sales.

Data Considerations

The following sections discuss data representation and training-data selection.

Data Representation

WebInsights analyzes websites as a binary classification problem. Websites are composed of many pages, only some of which are relevant to the segment in question. We follow the Standard Multiple Instance assumption, where the problem domain is a set of pages, and if one instance in the set is positive, then the whole set is positive.² After this filtering, we can treat the problem as a page, not binary, classification problem.

We programmed WebInsights to treat the web domain/problem as a standard text-classification problem. We chose to represent each document with the tf-idf (term frequency-inverse document frequency) for representation of its unigram, bigrams, and trigrams. The tf-idf weight is a statistical measure used to evaluate how important a word is to a document in a collection. The tf-idf weighting scheme is useful in scoring and ranking a document's relevance given a user query. To reduce computation time and memory footprint, we ignored rare n-grams—otherwise, our feature space would reach about 100 million features.

Sometimes sales and marketing users want to identify companies that match very few Intel resellers. For this, we used a condition that an n-gram must appear on the website of at least 0.5 percent of our resellers. We found that filtering by domains is more effective than filtering by pages, as it removes more

We programmed WebInsights to treat the web-domain problem as a standard text-classification problem.

² Standard Multiple Instance learning is a variant of inductive machine learning where each learning example contains a bag of instances instead of a single feature vector.

terms without damaging the model's ability to generalize. After filtering, the pages are represented by sparse vectors of about 50,000 to 100,000 n-grams depending upon the number of websites in the geography of interest and the language of the websites.

Data Selection for Training

Sales and marketing users typically have time to label only a few pages to define the segment. In most cases, only a small percentage (1 to 10 percent) of the resellers can be mapped to the chosen segment. If we were to randomly select pages for the user to classify, we would risk the scenario where none of the pages is labeled as relevant, thus preventing the system from learning a classifier.

The selection of which pages to label is therefore critical. Using a search query as a first step enables us to define two populations within our resellers: either the reseller's page was returned by the query and is thus somewhat relevant to the segment, or the partner's page was not returned by the query and is thus probably not relevant to the segment.

Users are given a number of pages that were found to be most relevant to the query, and this labeling helps the model refine the segment that was already roughly defined by the query. This dual approach of query discovery followed by document tagging leads to a better classification model when little training data is available. Since we do not want to spend time tagging pages that are not relevant, we randomly select a number of pages that were not returned by the query and automatically assign them the negative label (not relevant to the segment). This approach is valid because most pages are not relevant to the segment, and if a page is not returned by the query, it only increases the likelihood that the page is not relevant.

This dual approach of query discovery followed by document tagging leads to a better classification model when little training data is available.

Classification with Imbalanced Data

The page-selection methodology our system uses has a drawback. It forces an a priori or inferred class distribution that is defined by the system rather than by the data. If we choose to give the system the same amount of negative pages as there are positive pages, then in a standard learning process we incur the same penalty for misses as false alarms.

On one hand, each negative page in the training represents a greater number of negative pages (compared to the positive pages) that do not appear in the training data. On the other hand, we are trying to represent a cohesive underlying semantic category with a very small number of positive pages. Thus, we trained the classification model with various a priori probabilities for the weight of the positive class.

We used a k-fold cross-validation test to decide which method was best. In a k-fold cross-validation test, the data is broken into k subsets where some subsets are used in training and other subsets are used to test the training set. The advantage of this method is that it matters less how the data gets divided or how large each test set is and how many trial results are averaged.

Classification Based on Little Data

The most standard technique to perform text classification is by learning support vector machine (SVM) models over the bag-of-words representation of the pages. We decided to use this method as the baseline for our system. However, the high dimensionality of the n-gram space makes it prone to overfitting because we have so little training data. In this case, a feature that is not relevant to the segment may receive a positive weight because it appeared in positive pages but not in any of the negative pages. This does not mean this feature is relevant, but the model does not pay any penalty in assigning it a positive weight.

To reduce the overfitting problem, we tested several approaches:

- **Use robust models, which are less sensitive to overfitting.** Some models, like regularized or ensemble models, are known to be less prone to overfitting.
- **Get more training data.** With more data, we increase the probability that irrelevant features would appear associated with both positive and negative pages.
- **Reduce feature selection.** By reducing the feature space to a smaller dimensionality, we increase the ability to train good discriminative models from a small amount of training data.

Approach 1: Use Robust Models

First we explored the SVM version of the model with a parameter C that controls the penalty incurred for misclassifying samples in the training data. If C is large, misclassification of samples will weigh heavily on the SVM optimization criteria; conversely, small values of C mean that little penalty is incurred for misclassifying samples. Small values of C lead to models with broader margins that are more robust to noisy features. The optimal value of C depends on the data itself and can be determined by cross-validation over the training data. This approach seemed promising to help reduce the overfitting by averaging the results of multiple SVM classifiers learned from the training data. This method also allows for discriminative features that are dominant, appearing in multiple positive pages, to have an impact on page labels.

Approach 2: Gather More Labeled Pages

One alternative to obtaining more data from users is to use a semisupervised learning approach to obtain additional labeled data. This self-learning approach uses our base model to predict the labels on unlabeled data. We then select a number of the pages predicted as positive and negative with the highest degree of confidence and add them to our training data. By repeating this process several times, we can increase the amount of labeled data and reduce overfitting. One modification we have to impose is to add only one page per web domain during the process; otherwise, the algorithm risks adding all pages from one domain, which might lead the classifier to overfit the domain.

Approach 3: Reduce Feature Selection

Yet another way to reduce overfitting would be to reduce the number of features prior to running the SVM. Instead of running the feature selection from the model, we use results from user search queries. This means we treat the query results as a classifier—the pages that match the query receive a positive label and those that do not, a negative label. The newly labeled pages are run through a feature-selection process. Though not entirely accurate, the query runs over all the data, which means we have a much larger set of labeled pages. This is training by combining both feature selection and acquisition of new data.

Model Transfer between Languages

Use of machine-translation technology would appear to offer a solution to our problem of cross-language text classification. Users can enter their query and review pages in their language of choice (source language), and the model is then learned on these pages. Next, we translate pages from other languages to the source language and apply the learned model. The problem with this approach is that translating all pages would require a very expensive system. Since we have a small amount of labeled data, we decided it was more cost effective to translate labeled pages to our model in each language separately.

Some solutions to the overfitting problem require using query text to perform feature selection. However, the quality of translation of single words is weaker than translating these same words in the context of a whole page. For instance, the Russian word “обучение” could be translated into English as “training” or “education,” which in the case of understanding a reseller’s business focus would reflect two completely different industry segments.

Because user-labeled positive pages are drawn from the most relevant pages of the query, there is a way to recover an approximation of the context from the translated pages. We applied the feature selection from a previous query. The query words need to satisfy two properties: the probability ratio between the positive and negative pages should be high, and the term frequency in the positive pages should be high. If we were to find the k features that maximize the multiplication of the term frequency and probability ratio, we would obtain a good approximation of the feature-selection query.

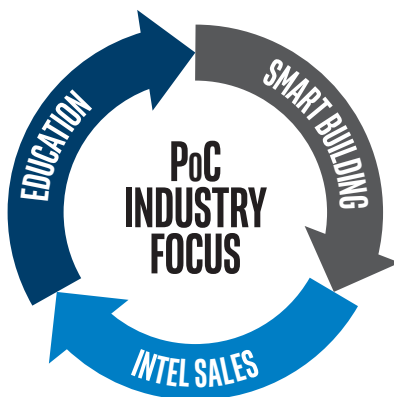
Initial Results from Our Proof of Concept

Our proof of concept (PoC) tested Intel resellers’ publicly available website content, and the results are reported for various industry segments of interest. We focused on three segments:

- Resellers selling to education institutions
- Resellers selling smart-building systems
- Intel's Sales Enablement Group

Because of the importance of the language-agnostic property of the algorithms, we reported the results on the Education segment on English websites and German websites. We refer to these as Education (English) and Education (German).

One difficulty was the overfitting issues that were created from having little training data. For the first test we chose a broad model, which improved our results. Because users prefer lists of companies with high-quality results as opposed to many results, we measured the quality of our model as the F1-score, which is a common method to average between both measures. For determining the optimal regularization parameter C for the regularized SVM, we performed a very wide search over values of the parameters ranging from 10^3 to $10^{(-7)}$. We selected the parameters that



provided the best F1-scores over 5-fold cross-validation tests. To learn the SVM's models, we used 100 estimators and drew samples from 75 percent of page replacements. We also drew 75 percent of the features to train each SVM.

As shown in Table 1, performing cross-validation tests to learn the regularization parameter C does help in reducing overfitting and marginally improves the F1-score. Using the ensemble method, on the other hand, improves the classification results and seems like a promising method. The main drawback of this method is that its computational complexity is much higher, and using it in an interactive system requires massive parallelization of the prediction phase.

The second test used various methods to alleviate overfitting by adding more data. For this test, we used the SVM with penalty optimization through 5-fold cross-validation as the baseline and then compared the following approaches:

- **1kNeg.** Using 1,000 sampled pages as negative out of all the pages that did not match the query (instead of 50 pages).
- **1kNeg bp.** Same as 1kNeg, but we force the model to weigh the penalty inversely to the actual sample distribution. We assume balanced prior class probabilities of both classes.
- **1kNeg ow.** Instead of weighting the penalty by rebalancing the prior class distribution in the training data, we optimize the class weights by using broad searches on the class weighting that gives the highest average F1-score over 5-fold cross-validation.
- **Semisupervised.** Results of the fourth iteration of self-training SVM with rebalancing of the priors (as in 1kNeg bp), where at each iteration we added five pages with the highest positive score and five pages with the lowest negative score.

As seen in Table 2, adding negative pages (1kNeg) results in a negative impact on the classification results, with lower F1-scores across all segments—the cost of the model having a bias towards negative pages becomes small compared to the impact of false-negative errors. Hence, the learning of the C parameter optimizes toward returning a large number of negative pages.

Table 1. Comparison of the Use of Regularized Support Vector Machine (SVM) and Bagging SVM to Hard-Margined SVM (highest result for each segment in bold)

	SVM	SVM with C	Bagging SVM
Education (English)	0.89	0.86	0.90
Education (German)	0.42	0.47	0.84
Smart Building (English)	0.79	0.83	0.79
Sales Enablement Group (English)	0.47	0.47	0.72
Overall	0.64	0.66	0.81

Table 2. F1-Scores Reported for Various Methods Used to Alleviate Overfitting by Adding More Data (highest result for each segment in bold)

	Baseline	1kNeg	1kNeg bp	1kNeg ow	Semisupervised
Education (English)	0.86	0.84	0.91	0.88	0.88
Education (German)	0.47	0.27	0.63	0.70	0.70
Smart Building (English)	0.83	0.69	0.84	0.82	0.81
Sales Enablement Group (English)	0.47	0.41	0.56	0.47	0.52
Overall	0.66	0.55	0.74	0.72	0.73

When we reweighted the optimization criteria to assume a balanced prior distribution between classes (1k Neg bp), the overall F1-score on our segments test sets dropped.

When we reweighted the optimization criteria to assume a balanced prior distribution between classes (1kNeg bp), the overall F1-score on our segments test sets dropped. By reweighting the penalty to balance the impact to false-positive and false-negative errors, we imply that both classes have equal prior probability. However, we know the positive class is much more rare than the negative class. By letting the algorithm try to find the optimal class reweighting (1kNeg bp), we expected improved results. We do see an improved F1-score for the Education (German) segment but not for the other vertical industry segments.

The results for the semisupervised self-training algorithm show that though it does improve the F1-scores compared to the baseline for three out of four segments, it is not the best methodology.

One of the problems we faced with self-training is that it did not converge, as shown in Table 3. After a number of iterations, we saw a decline in F1-scores. This led us to conclude that this approach would require more research before it could be usable.

In a final attempt to reduce the amount of overfitting, we tested the possibility of performing feature selection. We first ran a classification test using various subsets of features to represent each page. These subsets are obtained by performing feature selection over our labeled training pages using chi-square (X2) or information-gain (IG) methods. The final number of features, n, kept in each subset is chosen from the set {5, 50, 500}. The classification model we used for this test (ow) is an SVM with penalty and class weight learned through broad searches (the same as 1kNeg ow, but without expanding the sampled negative training data to 1,000 pages).

Table 4 shows that the IG method returned worse results than our baseline, ow. When we performed the feature selection using X2, we got better results than our baseline.

Table 3. F1-Scores of a Self-Training, Semisupervised Algorithm across Iterations

	1	2	3	4	5	6	7	8	9
Education (English)	0.67	0.70	0.71	0.71	0.72	0.72	0.72	0.76	0.72
Education (German)	0.83	0.83	0.83	0.82	0.81	0.81	0.78	0.85	0.85
Smart Building (English)	0.88	0.88	0.89	0.88	0.88	0.87	0.86	0.86	0.86
Sales Enablement Group (English)	0.85	0.85	0.84	0.82	0.52	0.61	0.61	0.61	0.61

Table 4. Comparison of Feature-Selection Methods from Labeled Data on Classification Accuracy (highest result for each segment in bold)

	ow	Information Gain (IG) n=500	IG n=50	X2 n=500	X2 n=50	X2 n=5
Education (English)	0.88	0.71	0.63	0.89	0.88	0.99
Education (German)	0.67	0.68	0.51	0.56	0.76	0.94
Smart Building (English)	0.83	0.78	0.39	0.85	0.88	0.73
Sales Enablement Group (English)	0.85	0.73	0.79	0.82	0.45	0.65
Overall	0.81	0.73	0.58	0.78	0.74	0.83

In the Education segment (both in English and German) we obtained the best results by using five features, compared to the Smart Building segment that obtained best results with 50 features. This is because the Smart Building segment is more diffuse, sometimes referring to industrial Internet of Things (IoT) companies or smart-home companies. Because of this segment diffusion, we needed more features to model it properly. When we look at the feature weights of the SVM for Sales Enablement Group tests, where X2 n=50 and X2 n=5, no feature was assigned negative weights, which meant that it overcompensated with high values for the penalty.

We assumed that by labeling the dataset with the query as a weak classifier, and then performing feature selection on it, we would receive better features for the modeling phase. We performed feature selection by using X2 and odds ratio (Odds). Table 5 shows that the query feature selection (QFS) did not improve results significantly over feature selection on the training data, apart from the case of QFS Odds n=1000, which gave better results than other methods.

As mentioned in the “[Model Transfer between Languages](#)” section, we were concerned about users’ ability to label pages in one language and use this information to label pages in another language. In our testing, we took the pages labeled in English for the Education segment and translated the text of these pages into German, Russian, and Polish. We then performed several classification tests, and these results are reported in Table 6.

We compared the quality of the translated text to the “native” Education segment in German to replicate a German user entering a query in German and labeling relevant pages. We measured the quality of feature selection (keeping n=5 features) with X2, Odds, and our heuristic approach of multiplying the probability ratio of the feature by its frequency (Freq*Ratio in Table 6). We compared test results to the original model and to a secondary feature selection with Odds and n=1000 on the output of this classification.

Table 5. Feature-Selection Results from the Query Using Either Odds Ratio (QFS Odds) or Chi Square (QFS X2) (highest result for each segment in bold)

	ow	QFS Odds n=1000	QFS X2 n=1000	QFS Odds n=50	QFS X2 n=50
Education (English)	0.88	0.97	0.87	0.97	0.81
Education (German)	0.67	0.96	0.84	0.87	0.92
Smart Building (English)	0.83	0.92	0.83	0.75	0.55
Sales Enablement Group (English)	0.85	0.73	0.52	0.63	0.67
Overall	0.81	0.90	0.77	0.81	0.74

Table 6. Classification Applied to Web Content in Native Language and after Machine Translation (highest result for each segment in bold)

	ow	Freq*Ratio n=5	Odds n=5	X2 n=5	Freq*Ratio n=5, Odds n=1000	Odds n=5, Odds n=1000	X2 n=5, Odds n=1000
English Native	0.84	0.98	0.84	0.99	0.98	0.95	0.98
German Native	0.75	0.74	0.68	0.94	0.78	0.82	0.84
German Translated	0.67	0.83	0.77	0.85	0.82	0.80	0.71
Polish Translated	0.52	0.82	0.76	0.61	0.80	0.70	0.61
Russian Translated	0.43	0.64	0.64	0.64	0.70	0.74	0.74
Overall	0.64	0.80	0.74	0.81	0.82	0.80	0.78

Table 6 provides the values from the SVM model with optimal penalty and class weights on the segments with translated training pages. The results were poor. To improve them we applied the principles of feature selection to the translation use case. When we applied very strong feature selection, keeping only the top 5 n-grams, we managed to improve results. When we performed a second iteration of learning with Odds methods and n=1000 features, we managed to improve the F1-score slightly more.

Results: Positive Business Impact

Since conducting our PoC, we have improved ResellerInsights by refining its algorithms, integrating additional data layers, and enhancing the tool's usability. More importantly, we are measuring the impact it has had on Intel's business.

Our first tests identified several resellers for whom we had minimal information relating to two industry-vertical segments in Intel's CRM database. In the first trained model, we identified 5x the number of qualified resellers in each segment. In the second trained and scored model, we identified a 20x increase in qualified resellers for each segment.

To measure the effectiveness of our lists, we created a controlled test by measuring the open and click-through rates of a communication sent to the following destinations:

- The ResellerInsights list.
- A first control group of "Active Resellers" engaged with Intel sales and marketing teams and participating in Intel product training. This is the type of reseller we want ResellerInsights to identify.
- A second control group of randomly selected companies with the same high-level properties as those from the ResellerInsights list.

As seen in Table 7, the ResellerInsights list generated 8x more qualified new resellers for the new industry segments than the number of Active Resellers in those new segments. In addition, the new resellers from the ResellerInsights list had open and click-through rates that were comparable to Active Resellers, and the click-through rate was 2.5x higher than that of our random control group that was used as a baseline. This demonstrates that ResellerInsights was able to identify a large number of resellers that are engaged as current "Active Resellers."

Table 7. Results of Measured Open Rates and Click-Through Rates for ResellerInsights, Active Resellers, and Two Control Groups

	Number of Resellers in New Segments	Open Rate	Click-Through Rate
ResellerInsights	8x accounts	43.6 percent	12.9 percent
Active Resellers	x accounts	51.5 percent	11.7 percent
Control Group	2x accounts	37.4 percent	5.1 percent

8X MORE QUALIFIED NEW RESELLERS
for new industry segments using ResellerInsights compared to current Active Resellers

Conclusion

Machine learning can help organizations quickly build models that enable intelligent solutions—solutions that create new revenue streams and differentiate predictive organizations from their competitors. As such, machine learning is becoming a business necessity. Whether exploring underground oil reserves or improving the safety of automobiles, organizations need a highly scalable, balanced, and robust infrastructure that speeds discovery and innovation while decreasing time to market.

Intel IT will use more and more machine learning to turn big data into deep insights and recommendations by getting machines to reason and prescribe a course of action in real time for our smart and connected world.

Our Reseller Knowledge Base mines data from resellers' website content to help Intel's sales and marketing teams better understand resellers' product and service offerings as they relate to Intel's new areas of focus, thereby supporting Intel's evolving business strategy. Our PoC showed that this tool was effective even when users invested little effort and that it worked across geographies and languages to identify relevant resellers. These results were possible because we solved intricate machine-learning problems such as learning from minimal labeled data and cross-lingual knowledge transfer.

Since the PoC, Intel sales and marketing teams have delivered communications to eight vertical industries across four geographies and in eight languages. The Reseller Knowledge Base immediately demonstrated value and was quickly adopted by staff. When we tracked resellers in the engagement chain after they were first contacted through ResellerInsights, we found that, in comparison with the rest of the sales pipeline, twice as many resellers advanced from leads to qualified leads. Their click-through rate for email newsletters is now three times higher, and they complete Intel training at a rate three times higher than the rest of the sales pipeline.

Though originally developed for targeted communication, we found the Reseller Knowledge Base platform can also be used as a tool for matchmaking between resellers. As Intel North America sales manager Shannon Poulin explains, "In the past, every time we met with one particular reseller, all we talked about were segments related to what we thought was their main business. We had no idea they could be a strategic reseller in our new vertical industry segments."

For more information on Intel IT best practices, visit intel.com/IT.

Receive objective and personalized advice from unbiased professionals at advisors.intel.com. Fill out a simple form and one of our experienced experts will contact you within 5 business days.

IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation:

- [Twitter](#)
- [#IntelIT](#)
- [LinkedIn](#)
- [IT Center Community](#)

Visit us today at intel.com/IT or contact your local Intel representative if you would like to learn more.

Related Content

Visit intel.com/IT to find content on related topics:

- [Intel IT's Data Center Strategy for Business Transformation paper](#)
- [How Intel's Data Lake Is Improving Sales and Marketing Decisions paper](#)
- [Extending Enterprise Business Intelligence and Big Data to the Cloud paper](#)
- [Integrated Analytics Platform Helps Sales and Marketing paper](#)
- [High Performance Computing for Silicon Design paper](#)



Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: [Learn About Intel® Processor Numbers](#).

THE INFORMATION PROVIDED IN THIS PAPER IS INTENDED TO BE GENERAL IN NATURE AND IS NOT SPECIFIC GUIDANCE. RECOMMENDATIONS (INCLUDING POTENTIAL COST SAVINGS) ARE BASED UPON INTEL'S EXPERIENCE AND ARE ESTIMATES ONLY. INTEL DOES NOT GUARANTEE OR WARRANT OTHERS WILL OBTAIN SIMILAR RESULTS.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS AND SERVICES. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS AND SERVICES INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel, the Intel logo, Xeon, and Intel Xeon Phi are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

*Other names and brands may be claimed as the property of others. Copyright © 2016 Intel Corporation. All rights reserved.

Printed in USA



1016/LMIN/KC/PDF