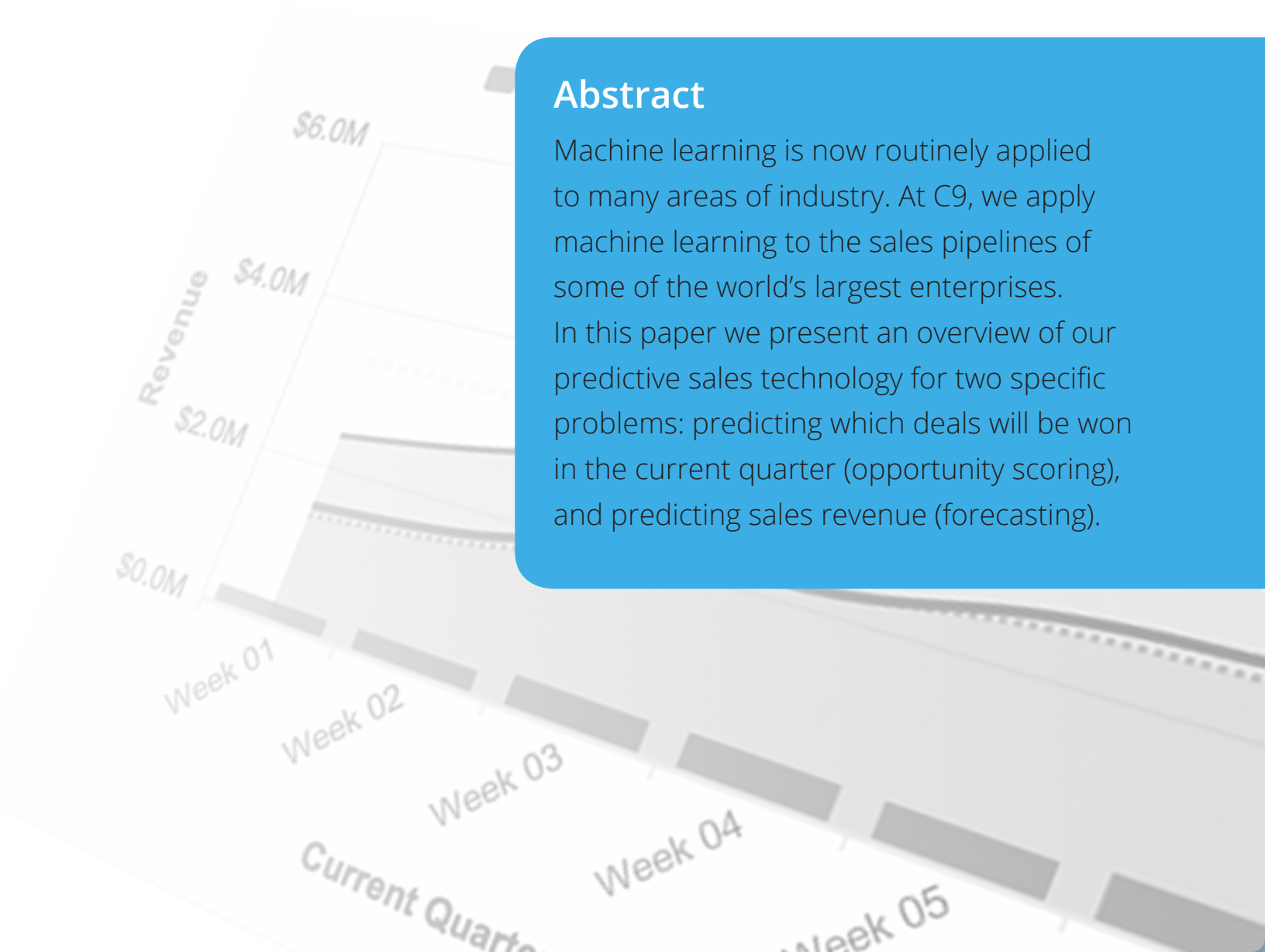
Applying Data Science to Sales Pipelines for Fun and Profit

Andy Twigg, *CTO, C9*

@lambdatwigg

Abstract

Machine learning is now routinely applied to many areas of industry. At C9, we apply machine learning to the sales pipelines of some of the world's largest enterprises.
In this paper we present an overview of our predictive sales technology for two specific problems: predicting which deals will be won in the current quarter (opportunity scoring), and predicting sales revenue (forecasting).



The need for an unbiased opinion

Traditionally, sales forecasts and pipelines as reported by sales teams are subject to subjective and emotional biases. These typically arise in several ways:

- **unrealistic targets:** sales reps are often required to have a certain coverage ratio (say 3x quota), and consequently they add low-quality deals they don't want to commit to.
- **happy ears:** sometimes, sales reps only hear the good news and not the bad, which biases their opinion of the opportunity. This is sometimes referred to as having 'happy ears'.
- **sandbagging:** sales reps' commission is often related to the fraction of quota they achieve, and this quota is adjusted based on their past performance. This gives an incentive to 'sandbag' – to delay reporting of deals that would otherwise be possible to close earlier.

The result is that pipelines contain spurious data and forecasts are missed. For many companies, this is a significant and common problem.

C9 helps eliminate these biases by applying data science to sales pipelines. Our models can typically identify winning opportunities over 45 days before closing, with over 80% accuracy. This helps sales teams in several ways:

- find opportunities that are promising but not committed (sandbagging)
- find opportunities that are committed but may be at risk (happy ears)
- gauge if the quality of the pipeline can support the current targets (unrealistic targets)
- produce a more accurate sales forecast

We will discuss several important questions that our predictive sales technology answers:

Opportunity Scoring

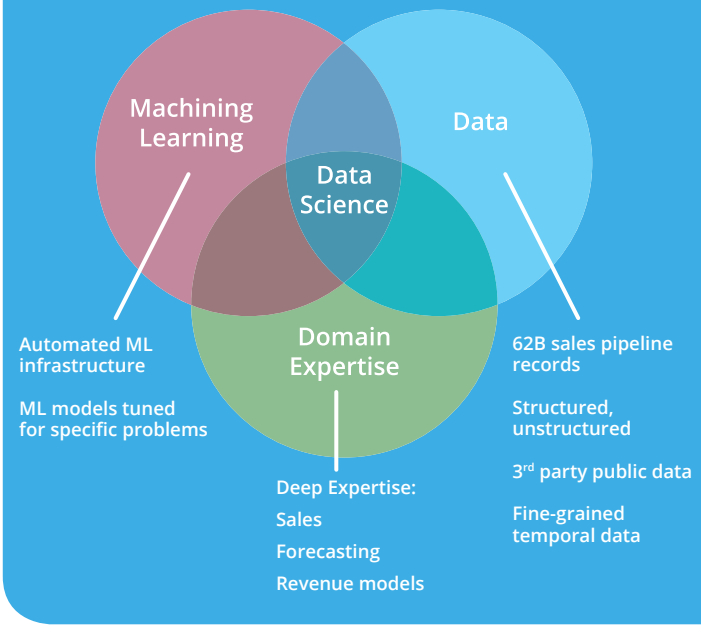
- What is the probability this opportunity will be won?
- What is the probability this opportunity will be won this quarter?
- How does this compare to sales team commits?
- What are the positive and negative factors influencing this deal?

Sales Forecasting

- How much are we predicted to close this quarter?

Putting the *data* in data science

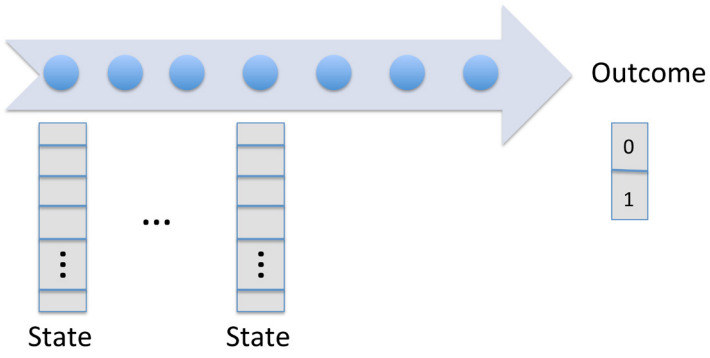
Data science is the intersection of three factors: machine learning expertise, domain expertise, and data. C9 guards the world's largest repository of fine-grained sales pipeline data – over 60 billion events, representing TBs of data. These record the history of sales opportunities from the first observation as a lead, to the last observation as a closed opportunity. Domain expertise enables us to transform and encode this data into a format that our machine learning algorithms can fully exploit. When combined with 3rd party and public data, this is one reason why C9 has the world's most accurate predictive sales models.



Opportunity scoring vs. lead scoring

Opportunity scoring and lead scoring have lots in common, but they are fundamentally different. A lead scoring model will typically take closed opportunities from a CRM system, extract some static information such as contact and account attributes associated with the opportunity (this will typically use around 5 fields such as domain name, contact name, position, email address), enrich this by consulting public APIs (this will typically add a large number of account-specific signals such as industry, technologies used, company size, job position of contact, etc.), and finally build a predictive model that can score new leads against this dataset.

In contrast, our opportunity scoring models make much heavier use of the temporal states throughout the opportunity's lifecycle, rather than the state when it is finally closed won/lost. Our models examine the (sometimes dozens of) changes in state as the opportunity progresses, and learn to identify patterns in intermediate states that are predictive of the final outcome (see the diagram below). This allows our models to consider a large number of temporal features derived by looking across multiple states, such as 'stage duration', 'amount change direction', 'number of pushes', 'how many times has this stage been visited previously' etc. that are not visible if the model was to only consider the final state of the opportunity.



In order to score an opportunity, the model looks at historic deals that have had similar behavior. In contrast to most predictive scoring systems, the model considers not just the current state but the past states, and how it progressed through those states. A good example of this is what happens if we look at the 'amount' field – our models often learn that, in later stages of the opportunity lifecycle, a fluctuating amount is a positive indicator for win, which intuitively represents some active negotiation. Similarly, the models consider the momentum the opportunity has, and the velocity with which it is progressing through the stages, and if it has revisited any previous stages. All these are possible positive or negative indicators for win.

Any machine learning model needs consistent, reliable inputs in order to build good models. Unfortunately, CRM systems are often subject to rapidly-changing schemas (for example, territory or hierarchy changes), and noisy data. In order to build good temporal features across multiple states, we need a way of reliably (and automatically) comparing two points in time, aligned with the same metadata, even if the metadata changed between those points. C9 is built on an advanced, proprietary temporal datastore that allows one to do exactly this.

Automated data pipelines

The entire model building and scoring process is automated, scalable and fault-tolerant. Our system can build a predictive model for a customer with almost no manual interaction or configuration.

The opportunity scoring engine starts by taking more than 1,000 raw signals per opportunity – this includes structured CRM data, unstructured text data (NLP), firmographic data including government sources (registration filings, credit unions, SEC filings), and other sources such as job postings, crunchbase, etc. Automated preprocessing and cleaning steps are applied to extract the important features containing signal, and to drop those features which are noisy (such as highly correlated subsets, fields with low variance, and so on).

Many potential models are constructed, with various 'hyperparameters', and each is automatically back-tested over the historical data to select the one with the highest accuracy. The models contain domain-specific features, such as 'number of days in stage', 'previous stages', 'number of times previously in this stage', and various metrics about the account and sales rep. Adding these features makes the models more powerful, and allows them to take advantage of the temporal nature of the opportunities. Each opportunity is re-scored every time it is updated, and the model itself is periodically rebuilt to take into account of recently-closed opportunities and new sales processes.

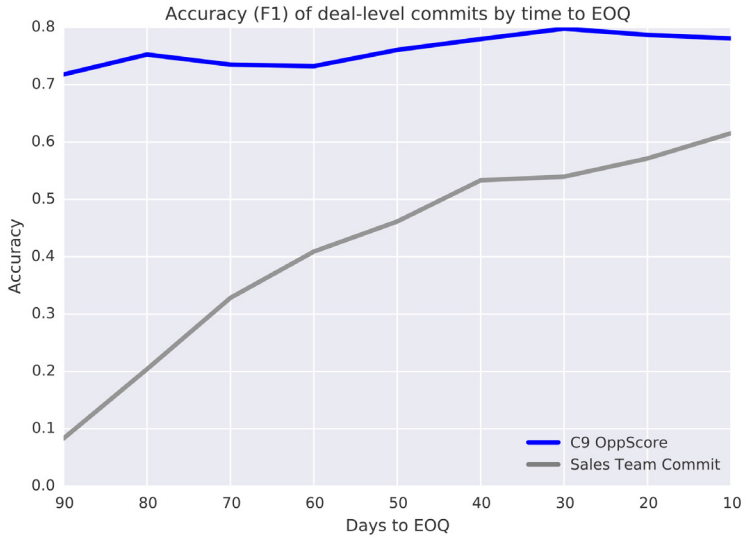
Our system actually maintains three separate predictive models, each tuned for the job of predicting a specific outcome:

- Will the deal win?
- Will the deal close in the current quarter?
- What are the main positive/negative influencers for this deal?

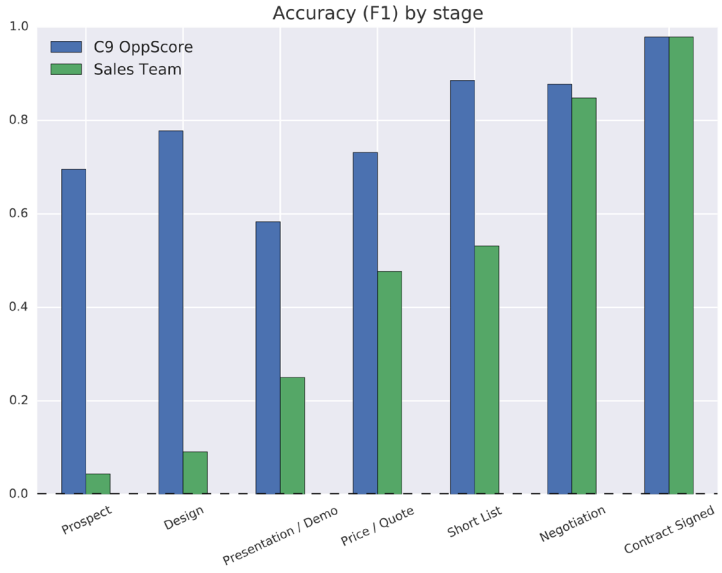
We discuss these submodels in more detail later.

Predictive models vs. sales team commits

The following chart compares the F1 accuracy of these predictions to the commit predictions made by the sales team, grouped by sales stage. Sales teams typically mark deals as 'committed' - we will treat this as a prediction that the deal will win in the current quarter (it will become clear later why we use F1 and not precision as our reported measure).



One can see that C9 is significantly better at predicting the outcome of opportunities earlier on than if we used the sales team commit. This translates into a significantly more accurate sales forecast early in the quarter. The figure below shows another example, this time split by stage - again, one can see that for the earlier stages, C9 is substantially more accurate than the commit flag.



Let's dig a little deeper into this performance gap. The table below shows the precision (fraction of predicted won opportunities that actually won) and recall (fraction of actually won opportunities that were correctly predicted) between the first observation of an opportunity and the last observation before it closed.

	First Observation			Last Observation		
	precision	recall	f1	precision	recall	f1
C9 scoring	0.65	0.86	0.74	0.75	0.93	0.83
Commit	0.70	0.07	0.13	0.87	0.45	0.59

One can see that both the commit and C9 precisions are reasonably high. Intuitively, this makes sense - sales reps tend to only commit deals that they are confident about, and our models also use the commit data as input. Since both precisions are high, we want a measure that takes into account both accuracy on committed deals (precision), and willingness to commit deals (recall). The F1 measure – the harmonic mean of recall and precision – provides exactly this. The table shows that the main difference between the C9 and Commit (sales team) accuracies is in the recall - the Commit recall starts at 7% and grows to 45%, while C9's recall starts at 86% and increases to over 90%. As a result, our F1 score is 5x better on the first observation than the sales team.

So C9 has approximately the same precision as the sales team commit, but with recall often several times better. In business terms, this means that when sales reps commit to a deal, it usually has a good chance of winning; but there are many won opportunities that don't get committed, particularly early on in the quarter. This is where C9's scoring can help most – the machine is willing to commit to deals much earlier than the rep. This allows managers to allocate resources and plan better, leading to less discounting.

Anatomy of an opportunity

It is illuminating to look at the behavior of a real opportunity. The figure below shows an opportunity as it progresses from start to finish, along with C9's predictions along the way. (The table only shows a small number of features for simplicity; the probabilities shown are generated by considering the full set of signals.)

Date	Stage	Amount	Days to EOQ	Days to close	Est days to close	Final outcome	Committed?	Pr(close in Q)	Pr(win)	Pred Win?	Pred Win in Q?
8/29	Qualify Opportunity	304128	33	303	115	1	0	13.8%	54.8%	**1**	**0**
9/28	Qualify Opportunity	304128	3	273	85	1	0	1.2%	56.0%	**1**	**0**
9/28	Discover Needs	304128	3	273	153	1	0	1.2%	59.1%	**1**	**0**
10/24	Discover Needs	432198	69	247	127	1	0	25.4%	59.2%	**1**	**0**
11/10	Discover Needs	645649	52	230	110	1	0	20.5%	59.4%	**1**	**0**
12/21	Discover Needs	645649	11	189	161	1	0	3.5%	58.2%	**1**	**0**
2/8	Discover Needs	384211	53	140	112	1	0	20.4%	57.5%	**1**	**0**
3/23	Negotiate	192959	10	97	44	1	1	13.3%	81.5%	**1**	**0**
4/19	Negotiate	192959	74	70	42	1	1	62.7%	81.3%	**1**	**1**
4/26	Negotiate	192959	67	63	58	1	1	48.5%	81.0%	**1**	**0**
5/9	Negotiate	192959	54	50	45	1	1	47.4%	81.7%	**1**	**0**
5/31	Negotiate	192959	32	28	8	1	1	84.8%	83.0%	**1**	**1**
6/9	Negotiate	192959	23	19	1	1	1	94.4%	83.1%	**1**	**1**
6/14	Negotiate	192959	18	14	6	1	1	72.4%	82.9%	**1**	**1**
6/23	Negotiate	192959	9	5	4	1	1	59.3%	83.0%	**1**	**1**
6/25	Closed/Won	192959	7	0	0	1	1	100.0%	100.0%	**1**	**1**

Immediately, one can see the following. The opportunity was eventually won ('final outcome'), but it took 303 days (each horizontal line denotes a sales quarter boundary), while the sales rep initially estimated it to close within 115 days. Every time the 'est days to close' measure increases, this indicates that the opportunity was pushed out.

The predictive model is predicting two things: the final outcome of the opportunity (win/lose), and the probability of closing in the current quarter (time horizon). The two columns Pr(win), Pr(close in Q) are the predicted probabilities of these events. The final two columns are binary indicators based on thresholding the probabilities -- for this example, the threshold is 0.5 (in practice, this is tuned depending on the customer's desire to trade-off false positives against false negatives).

One can see that the model correctly predicted the opportunity to eventually win, and the probability increased throughout the lifetime of the opportunity. In particular, when the opportunity changes stage to 'negotiate' and is committed by the sales team, this is a strong signal to the model, and Pr(win) jumps from 57% to 81%. The model also correctly predicted the opportunity to close in the final quarter. In contrast, the sales rep committed the opportunity with only 10 days remaining in the third quarter.

An interesting event occurs into the fourth quarter - the opportunity is pushed out (est days to close goes from 42 to 58), which causes the model to predict that it won't close in the current quarter. The opportunity is subsequently pulled back in, and the model again predicts it to close in the quarter.

Technical details on predictive models

The system maintains three models in parallel, each trained for a different job.

Win/loss model: predicting outcomes

The overall win/loss model attempts to directly predict Pr(win) for an opportunity, independent of the time horizon. This is computed by feeding the data into a calibrated random forest classifier, which provides state-of-the-art accuracy. Many potential models are built with varying parameters, and each is automatically cross-validated against historic data to determine the best model for the customer's data and sales process.

Duration model: predicting time-to-close

A unique feature of C9's machine learning, the 'duration model' predicts the probability an opportunity will close in the current quarter. This is done by a technique known as Poisson regression – we assume that, in its current state, an opportunity has some fixed probability of closing each day. Integrating the corresponding exponential distribution gives the quantity Pr(close < t) for some time horizon t. An optimization we make here is to only train on won opportunities – thus, the output probabilities are conditionals - as this significantly improves the quality of the prediction (we are not interested in predicting close dates for lost opportunities). We are particularly interested in the time horizon t_q, i.e. the time until the end of the current quarter. Combining these conditionals with the win/loss model, we can compute

$$Pr(win) \times Pr(close < t_q \mid win) = Pr(close < t_q \text{ AND } win),$$

the probability that a given opportunity will be closed-won in the current quarter. This description omits many small technical tricks – each improves the quality of the model by a few percent, but they quickly add up to produce a highly accurate model.

Influencer model: surfacing important drivers

A common trade-off in machine learning models is between predictive power and interpretability, i.e. the ease of explaining its predictions. Powerful ensemble models such as random forests or gradient boosted trees are a classical example - they often achieve excellent precision and recall, but are generally difficult to interpret.

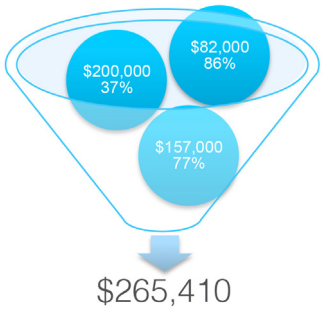
We would like to report, for each opportunity in its current state, which features contribute positively or negatively to its current win prediction. We provide this information as follows. We use our domain specific knowledge to generate features that capture key drivers of the sales process like momentum, pricing, and contact strength. In parallel to the Pr(win) model above, we compute a GLM model, such as an elastic net[1], on the same data with the same target variable. We then use the standardized coefficients and importances to determine, for each opportunity, the specific features and values that influenced the prediction both positively and negatively.

Predictive sales forecasting

In this section, we describe how we take opportunity-level predictive models one step further, to produce an aggregate predictive forecasting model. In order to do this, we first review two well-known forecasting approaches - bottom-up and top-down – and then show how we combine them into a 'hybrid' model.

Bottom-up forecasting

One can produce a 'bottom-up' forecast from the active open opportunities in the sales pipeline by summing all the amounts in the pipeline, weighted by their predictive scores – as in the example at right.



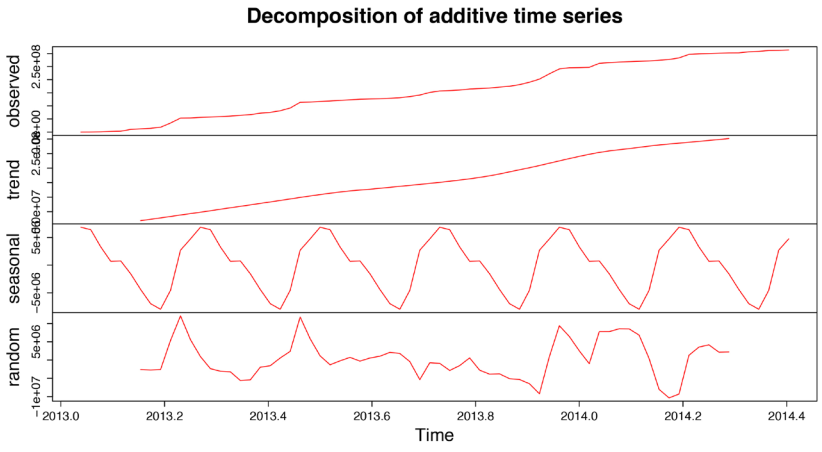
Bottom-up forecasting aggregates the total value in the pipeline, weighted by probability of win in the current quarter

This method has the advantage that it takes into account the quality and quantity of the current pipeline. The disadvantage is that, while it predicts accurately towards the end of the quarter, it doesn't take into account deals not yet in the pipeline, so it performs poorly near the start of the quarter, especially in cases where the sales cycle length is less than the prediction period (a sales quarter).

Top-down forecasting

Traditional time-series forecasting methods such as **ARIMA**[2] can be trained on the historical aggregate revenue from past quarters, and given the current revenue, can be used to predict the final revenue in the quarter (we call these 'top-down' in contrast to the bottom-up method). This is indeed a very common use case for these methods. The disadvantage of the techniques, however, is that they do not take into account the quality of the current sales pipeline; they only consider the past revenue trend and project this into the future.

This figure shows a typical decomposition of a revenue time series into three components:



1) **a trend component** – this typically captures long-term trends such as growth
2) **a seasonal component** – this typically captures cyclical trends such as increased growth at the end of each quarter, seasonal impacts, holidays, etc.
3) **a random component** – this includes whatever is left over from the previous two components. One can usually think of this component as the fluctuations in the data that the model cannot explain.

The idea behind C9's predictive forecasting approach is to use the bottom-up forecasts generated by the opportunity scores to better explain the random component, and thus improve the accuracy of this model.

‘Hybrid’ forecasting

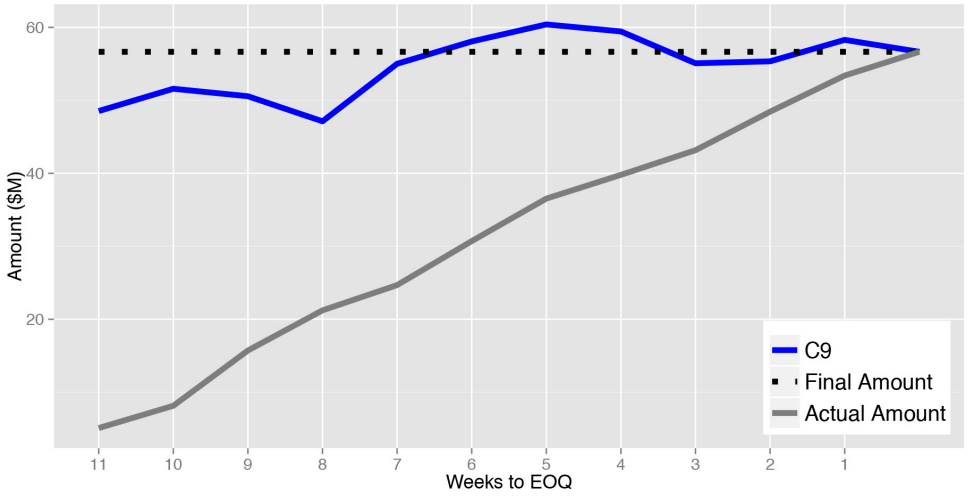
By combining the top-down and bottom-up approaches, we arrive at C9’s ‘hybrid’ predictive forecasting model. This offers the best of both worlds. The model operates by creating exogenous variables representing the ‘quality’ of the current pipeline and using them in a time-series formulation, in a manner similar to the **ARIMAX** model[3].

This method allows the model to account for the following:

- the outlook of deals currently in the pipeline
- the state of pipeline compared to how it looked in the previous quarter
- the trajectory of deals not in pipeline, bluebirds, seasonality and cyclicality (as captured by the top-down model)

The example below shows the relative error (NMAE) of the predicted final amount, compared to the actual final amount closed at the end of the quarter. Our forecast is usually more than 95% accurate, averaged over the course of the quarter.

This hybrid forecasting algorithm is somewhat more advanced than most competitive solutions that simply extrapolate previous sales trends, and thus do not take into account the quality of the current sales pipeline.



Conclusion

C9 has the world’s most accurate predictive sales models, based on the world’s largest repository of fine-grained historical sales data, built, tuned and optimized for individual customers. These opportunity-level scoring models are then leveraged to produce the C9 hybrid forecast model – which delivers the most accurate predictive forecasts, taking into account both previous trends and seasonal variation, and the quality of the sales pipeline. Together, these give sales teams a reliable and objective view into their sales pipelines and forecasts that they never previously had.

1 - http://en.wikipedia.org/wiki/Elastic_net_regularization
2 - http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
3 - http://robjhyndman.com/hyndsight/arimax/

C9 Inc.
177 Bovet Road, Suite 520
San Mateo, CA 94402

Call us: (650) 561-7855
Email Us: info@c9inc.com
www.c9inc.com