Closing the Deal: Predicting Sales Across Companies

Devini Senaratna, Benjamin Bernstein, Daniel Post
Stanford University

Introduction

In this project we focused on answering three questions about sales opportunities: 1) For how much will the deal close? 2) Will the deal close? 3) When will the deal close? Answering these questions helps a company allocate resources, precisely forecast outcomes, maximize revenue, and set quotas accurately. The uniqueness of this project is in developing models to combine data across companies.

Our data came from Roam Analytics, a company that collects and analyzes sales data for several companies. Our project builds on current methodologies by combining the data for the companies instead of modeling them separately. The advantage of combining the data is that when Roam acquires a new client, it might be difficult to make accurate predictions about sales opportunities until enough data has been gathered. In this setting we can leverage Roam's existing database of sales information across all of the companies to make predictions for the new client.

Data

The data was obtained from three companies. Company A is a large internet storage organization with over 100,000 clients. Company B is an IT management software and solutions company with over 10,000 clients. Company C is an independent technology and research company with about 2,500 clients.

The three variables we tried to predict were sales dollar amount, a binary win/loss variable, and a binary close within 30 days variable. We removed outliers when the amount of sale was less than \$10K and more than \$300K. Doing this left us with 22270 training samples and 6865 test samples. Our predictors consisted of event unigrams, bigrams, and trigrams. These tuples were designed using features such as the probability that the salesperson expects the deal to close, the predicted sales amount change, email activity, and expected close date change. An example of a tuple is: probability of close changed from 10% to 20%, there was an email interaction with the client, and the expected amount of sale changed from \$15k to \$20k. Roam Analytics had already found these features to correlate the best with sales prediction. In the end we had 823 variables from which to choose.

One problem we encountered was the inconsistencies between variables across companies. For example, companies give each sale a stage name to measure its progress and categorize the sale. However, they use a different number of stages and stage names so we could not merge them. If the n-grams did not have the same stage names, then we dropped them from the merged dataset. Not all information about the stages was lost because we included the probability of sale, which is associated with stage names. The probability groups were consistent across companies so we could include them in the merged dataset.

Descriptive Statistics

First, we investigated the three dependent variables starting with the sale amount. We observed that the distribution of the sale amount appears to be exponential. To correct for normality we applied the log transform and the power transform. Looking at the R^2 values of the Q-Q plots (Figure 1) we see that both transforms improved normality, with the power transform being slightly better. Also, looking at the bar plots of the binary dependent variables we see class imbalances, which we can address in our modeling (Figure 2).

Two proposed methods to develop better features were: (1) to create new uncorrelated features using PCA and (2) to use cluster labels as a predictor of our dependent variables. Both these methods were unsuccessful, as the first 10 principal components did not explain the variation of the features well (a total explained variance of 0.46, as seen in Figure 3) and k-means clustering did not provide a very good separation between cluster groups.

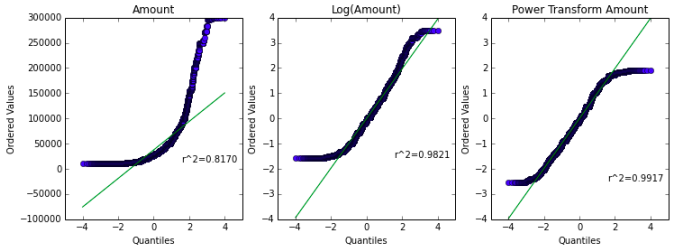

Figure 1: Q-Q plots of amount-of-sale, log(amount-of-sale), and the Power Transformation

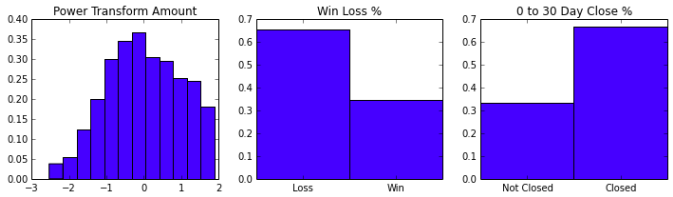

Figure 2: Distributions of the three dependent variables

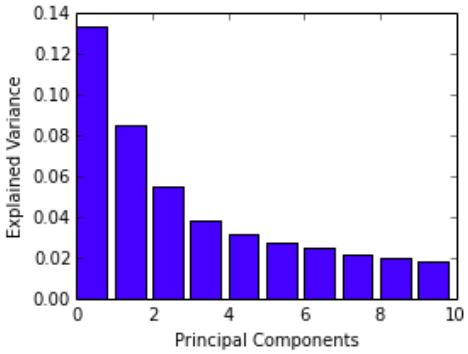

Figure 3: Scree-plot obtained using PCA

Modeling

Sales Amount

Starting with the sales amount problem, we broke up the task into fast and slow computational models. Due to the high dimensionality of our data, we decided to focus on Lasso regression as a means of fast computation, hoping to have many of our coefficients shrunken to zero. We included Ridge regression and a multi-level model (MLM) as alternatives, which are also computationally fast. The results of all models were similar; the learning curves show high bias and low variance (Figures 4 and 5). The errors on the test set for Lasso show a high mean squared error at 0.87 that is centered but not very concentrated on zero (Figure 6). The errors on the test set for MLM looked very similar to Lasso and had a mean squared error of 0.98. It was clear that we could do better by addressing the bias in the model by adding variables. We proceeded to use a Random Forest regression as our slow computation model to solve the bias problem.

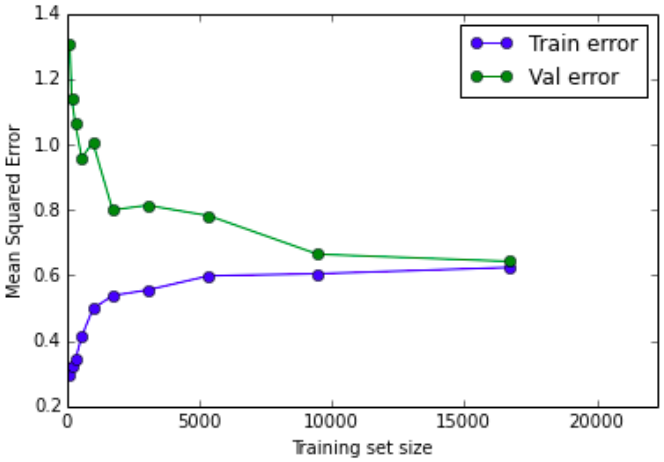

Figure 4: Training and validation-set error-curves for predicting Sales Amount using Lasso Regression

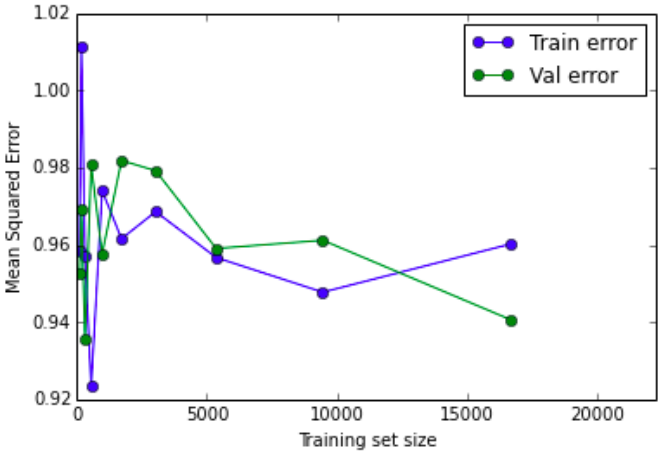

Figure 5: Training and validation-set error-curves for predicting Sales Amount using Ridge Regression

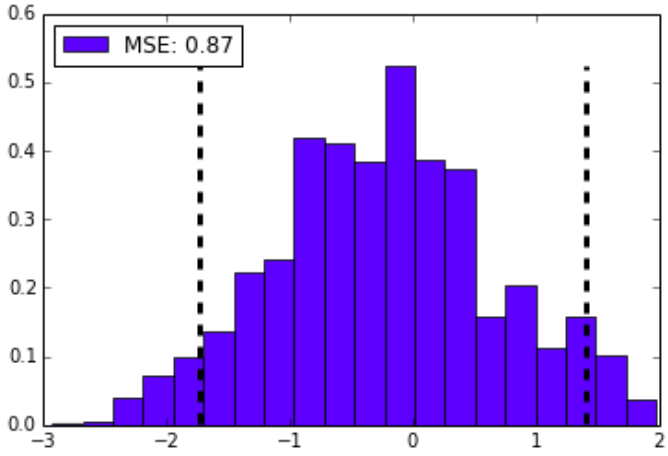

Figure 6: Histogram for the squared difference between the predicted, transformed sales amount using Lasso regression and the actual sales amount (based on the test-set)

The learning curve for Random Forest regression (Figure 7) shows a decrease of bias and almost no increase in variance, which was our goal. Finally,

looking at the evaluation on the test set showed a mean squared error of 0.26 that was both centered and highly concentrated around zero (Figure 8).

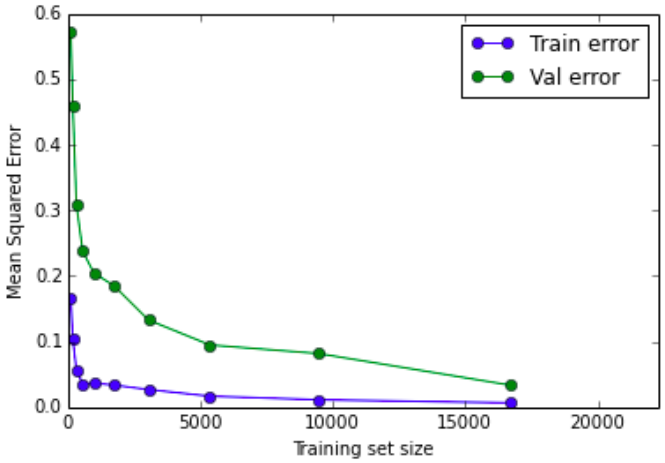

Figure 7: Training and validation-set error-curves for predicting Sales Amount using Random Forests

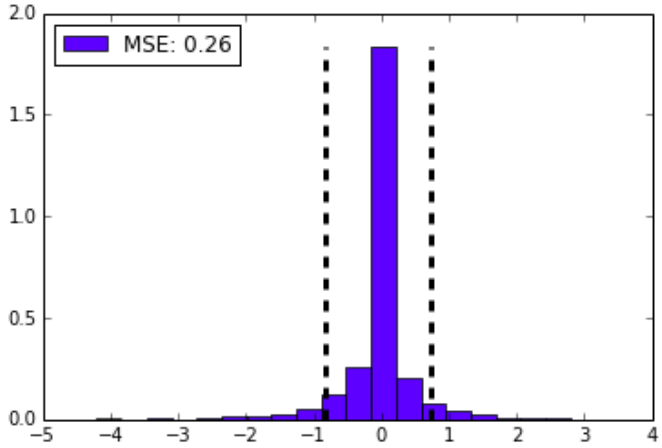

Figure 8: Histogram for the squared difference between the predicted, transformed sales amount using Random Forests and the actual sales amount (based on the test-set)

To understand the results of the improved model we converted our predictions back to dollars and compared them to the untransformed test values. We saw a large skew in the errors when in dollars since the root mean squared errors are all much larger than the root median squared errors. Also, we observed that the training data size was inversely correlated with the model error. That is, companies with more training data had smaller errors. Finally, we plotted the predicted dollar amounts against the true dollar amounts and observed a correlation coefficient of 0.81, indicating a satisfactory model (Figure 9).

Table 1: Transformed dollar-amount root mean squared errors and root median squared errors by company.

Company Name	Training Data Size	RMeanSE	RMedianSE
A	1982	43651.91	7767.46
B	5904	35117.84	4397.89
C	14384	19173.47	1141.44
Total	22270	22056.65	1440.72

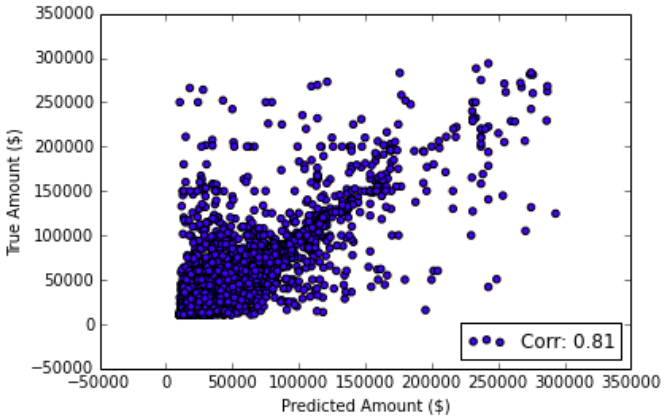

Figure 9: Scatter-plot of predicted sales amounts against the true sales amount (based on the validation-set)

Win-Loss and 0 to 30 Day Close:

Unlike sales amount, both win-loss status of a prospective sale and if a sale closed within 30 days could not be modeled satisfactorily. Similar to the amount modeling procedure we tried fast and slow computational models. We started with L1 Logistic and Naïve Bayes and again noticed high bias from looking at the learning curves. To address this problem we attempted to use a support vector machine (SVM) with the kernel trick for a higher dimensional model. However, in this case the bias was not reduced. One possible explanation for the low F1 scores (Van-Rijsbergen, 1979) is that our n-grams dropped the stage names and that caused us to lose features with high predictive power. Another possible explanation is that the distributions of the conditional dependent variables were different in the training from the testing data sets. It seemed that there was a systematic change in the proportions of wins and losses after 12/01/2012 (our cut off for creating the training and testing data sets).

We produced confusion matrices below to present the results. The confusion matrices for win-loss showed that the models predicted losses well but failed to predict wins (Table 2). The confusion matrices for 0 to 30 day close showed the models predicted closed well, but were unable to predict not-closed (Table 3).

Table 2: Confusion matrix information for win-loss status

Type	F1 Score	True Negative	False Positive	False Negative	True Positive
Logistic	0.204	417	5705	12	731
SVM	0.204	473	5649	14	729

Table 3: Confusion matrix information for 0-30 day win-loss status

Type	F1 Score	True Negative	False Positive	False Negative	True Positive
Logistic	0.044	5620	135	1082	28
Naive Bayes	0.133	4968	787	975	135
SVM	0.183	4856	899	908	202

Conclusion

In order to predict sales amount ridge regression, lasso regression, multilevel models and random forests were attempted. By observing the learning curves for ridge and lasso regression, it was evident that these models had a high bias. The solution for this problem was to use random forests, which provided a highly satisfactory model. Predictions on the test data for companies A, B, and C gave a root median squared error of less than $1500 overall and less than $8000 for individual companies.

The other two dependent variables, win-loss status and 0-30 day closure status, could not be predicted well. Many types of models including logistic regression, Naïve Bayes, and support vector machines were attempted along with diagnostics from confusion matrices, Receiver Operating Characteristic (ROC) curves, and learning curves.

Roam's prior analysis on individual companies provided satisfactory models for win-loss and 0-30 day closure. Therefore, it is likely that the features based on the stage names are vital for predicting these two dependent variables. To elaborate, in the process of combining the data we excluded stage names as we generated the n-grams and used just the change in probability. Because there are multiple stage names for each change in probability, removing the stage names caused us to lose predictive power. In the future, we would suggest that Roam creates a general mapping across stages so we can include them when we merge features across companies.

When onboarding a new client, it will take time to gather enough data to start making accurate predictions just based on that client's own sales history. We can use the results of this project as a first step to making sales predictions and providing an initial assessment for new clients.

Acknowledgments

Special thanks to Andrew Maas for introducing us to the project, Kevin Reschke for helping with feature engineering, and Joe Barefoot for database management.

References

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag.

James, G., Witten, D. Hastie, T., and Tibshirani, R (2013), *An Introduction to Statistical Learning with Applications in R*, Springer.

Van-Rijsbergen, C. J. (1979). *Information Retrieval*, Butterworth.