Propensity of Contract Renewals

Himanshu Shekhar (hshekhar@stanford.edu) - Stanford University

I. Introduction

Multinational technology firm develops, manufactures, and sells networking hardware, telecommunications equipment, and other high-technology services and products. Major portion of their revenue is driven from its existing customer base through the services (maintenance, troubleshooting, installation etc.) they sell along with the product. That means even a modest churn rate can cause a dent in your revenue. However, these services are provided only for a limited timeframe(6months/12months) after which customers have to renew their contract in order to continue receiving the services. Hence they rely heavily upon contract renewals for their ongoing business due to subscription based nature of the business model and they use this information to accurately predict their revenue. It is very critical for the businesses to accurately predict the likelihood of contract renewal and also understand the underlying drivers which impact the renewal propensity score. Quantifying the impact of different factors/drivers would help business to understand what parameters/levers they can tweak to maximize the likelihood and use this information to their advantage. Idea/objective is to build a probabilistic model which will provide the renewal propensity score for each contract well in advance of the expiration date so that sales team can prioritize the set of contracts which have low propensity and also provide them with details on different factors so as to maximize the renewal conversion.

II. Data

Raw dataset contained more than 50 fields for each contract line. However, not all of the fields are intuitively useful for the learning model, such as the contract ID, product key etc, and thus I removed such fields. I have included some of the attributes which are relevant for the model which focuses mostly around customers engagement/transaction with the technology service provider like customer market segment, geography, of contracts held, of same product type installed, wallet size, wallet share, business vertical, tenure of association, channel of purchase, history of non-renewal, tenure of association, service subgroup, length of contract, discount, quote generation time (of days before/after expiry), product type, of days to end of sale, of days to last day of support, list price, of co-terminating contract lines, # of service requests/complaints raised, resolution time to service requests, type of service, region, service level agreement. Data pre-processing was done to ensure that we dont have missing values, outliers. Categorical features, such as market segment, channel of purchase, were expanded into Boolean columns, one column for each distinct value that the feature could take. Also, some of the other features like product family, product type were grouped together depending upon the renewal rate observed across them and then converted into Boolean columns. To label the dataset, I classified any contract that expired as negative(0) examples, while I classified any contract that renewed as positive (1) examples and this variable would be used as dependent variable in the model

III. Feature expansion

Lot of important information is captured in the case notes (interaction between customer and the service provider while resolving for the service request) which is in free form text which could help us understand customers sentiment towards the service provider and could also help us understand the variation in renewal propensity. To extract information out of these case notes, I used text mining (specifically TF-IDF). TF-IDF allows us to determine any important words in each case note while taking into account the number of times a word appears in the corpus so that words that occur frequently in general are weighted lower.

$$tf(t,d) = f_{t,d}$$

$$idf(t,D) = log\frac{N}{|\{d \in D : t \in d\}|}$$

$$tfidf(i,d,D) = tf(t,d) * idf(t,D)$$

$tf(t,d)$, or term frequency, is equal to the number of times term t occurs in document d. $idf(t,D)$, or inverse document frequency, is equal to the log of the total number of documents divided by the number of documents d in the set of all documents D that contains the term t. The TF-IDF score is the product of the term frequency and the inverse document frequency. Before running TF-IDF on the case notes, I removed any punctuation and html tags, tokenized and stemmed the text using the Porter stemming algorithm, then removed any English stop words.

I split the corpus into two groups, one for renewed contracts and one for expired contracts. For each group, I determined the unique words across all case notes, calculated the TF-IDF score for each word for case note, then summed up the scores. Since I am looking for terms that occur in either renewed or expired contracts but not both, I normalized the TF-IDF scores for each group and chose words with the highest absolute difference in the normalized TF-IDF scores between the two groups. The top words were response time, advertising errors, in-person interactions, billing issues, customer service. I then created binary features for each of the words indicating whether that word is present in the case notes

IV. Method

All learning algorithms considered for use for this project falls under the category of supervised learning methods for classification. Multiple models (logistic regression Support Vector Machine) were tested with varying parameters to find which algorithm results in the highest performance rates. A detailed description of each algorithm, as well as its strengths and weaknesses with respect to this project, would be done in the later sections. I implemented all algorithms in Python, using the sci-kit learn library.

Accuracy alone will not help us assess the true performance of the model. Along with accuracy, I will also look at other metrics like sensitivity, defined as:

$$sensitivity = \frac{TP}{TP + FN}$$

where TP is the number of true positives and FN is the number of false negatives. Sensitivity is the fraction of contracts that are actually positive (renewed contracts) that were predicted as positive by the model. Since, we need to have good predictability for both renewed expired contracts, we also need to look at specificity, defined as:

$$specificity = \frac{TN}{TN + FP}$$

where TN is the number of true negatives and FP is the number of false positives. Specificity is the fraction of loans that are actually negative (expired contracts) that were predicted as negative by the model. In order to combine both sensitivity and specificity, we will use G-mean:

$$G = \sqrt{sensitivity * specificity}$$

Also, for completion of performance metrics, I also looked at accuracy and precision:

$$accuracy = \frac{TN + TP}{N}$$

$$precision = \frac{TP}{TP + FP}$$

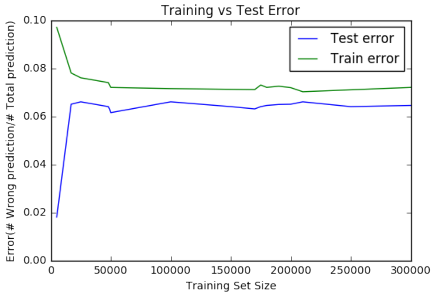
To establish the performance, I train each model on 70% of data randomly selected and test them on the remaining 30% of the data.

V. Logistic Regression

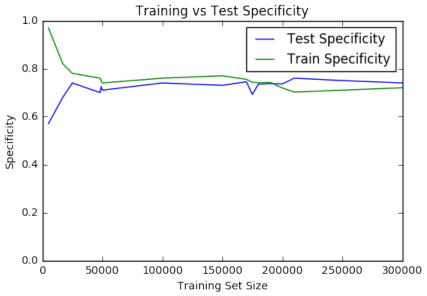
I try modeling with logistic regression with Newtons method to learn more about the data features and get the basic performance of the prediction. To understand predictive contribution from each data features, I trial-trained with a logistic classification on all features and got rid of all the factors/variables where the p value is greater than .05 so that we are 95% confident that upon repeated trials, 95% of the confidence intervals (CI) would include the ’true’ population odds ratio. If the CI includes one, we’d fail to reject the null hypothesis that a particular regression coefficient equals zero and the odds ratio equals one, given the other predictors are in the model. An advantage of CI is that it is illustrative; it provides information on where the ”true” parameter may lie and the precision of the point estimate for the odds ratio. The training and test converges to an optimal solution within 11 iterations, and overall I reached a test accuracy of 89.7% and a test specificity of 74.7%

A. Bias vs. variance

To see how the logistic model can be further improved, I ran a diagnostic by different sample size:



Training vs. Test errors



Training vs. Test specificity

The test and training error converged quickly with a sample size $>= 50,000$, and we see that we may have a high bias problem as increasing sample size still resulted in a $> 5\%$ test error. From the sensitivity chart, however, we see that sensitivity fluctuates with additional sample size, suggesting that the renewal prediction might potentially benefit from filtering on existing features even though test error has stabilized.

B. *Feature Selection*

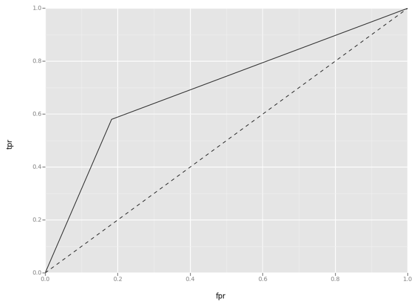
I ran the variation inflation factor (VIF) test to ensure there is no multicollinearity in the data and either dropped the features or created a derived metric by combining the features wherever the VIF value was more than 5. Further I found that the quote variable (categorical variable which captures whether quote was generated before the expiry) was highly correlated with the renewal flag and was one of the major contributor in explaining the dependent (renewal flag) variable. I achieved a high accuracy of 94% and specificity of 80% with the quote variable in the model. However, quote is generated 30 days before the expiry date only for contracts which have certain minimum $ value due to resource constraints and most of these contracts renew, so I removed this feature from the modelling dataset as it was explaining major portion of the variation in the dataset (i.e. suppressing beta estimates of other features) and also because it provided a shorter opportunity window of 30 days to persuade a customer. Further Using ablative analysis on the logistic model, I found that for prediction on renewal flag, the number of days to last day of support is the most predicative of all features, followed by service type; while pricing has negative impact as the number was subject to sporadic adjustment from the service provider, and response from customer surveys , where there are a lot of missing values, would worsen the renewal prediction. After I filtered out features that decreased the test specificity, such as the discount rate, survey responses, I managed to bump specificity from 74.7% to 76.1% without hurting overall test accuracy:

TABLE I. PERFORMANCE OF LOGISTIC MODEL WITH FEATURE SELECTION

Num Newton	Accu	Prec	Sens	Spec	G-mean
40 iterations	91.4	95.1	94.2	76.1	84.6

C. *Receiver Operating Characteristic (ROC)*

The position of the cut-off determines the number of true positives, true negatives, false positives, and false negatives. As you increase your sensitivity (true positives) and can identify more cases with a certain condition, you also sacrifice accuracy on identifying those without the condition (specificity). In this case we would want to maximize specificity as we are more interested in identifying the contracts which will not renew (true negatives) so I plotted a ROC curve to get a better sense of the threshold I should use.ROC curves feature true positive rate on the Y axis, and false positive rate on the X axis. This means that the top left corner of the plot is the ideal point - a false positive rate of zero, and a true positive rate of one. This is not very realistic, but it does mean that a larger area under the curve (AUC) is usually better - I obtained an AUC of 0.76



Receiver Operating Curve

VI. Support Vector Machine

Since the training data is likely not linearly separable, and not guaranteed to be separable even in higher-dimensional feature spaces, we will use L1 regularization (soft margin SVM). For training data points $(x^{(i)}, y^{(i)})$, the model is the result of the optimization:

$$min_{\gamma,\omega,b}\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t.y^{(i)}(\omega^T x^{(i)} + b) >= 1 - \xi_i, \xi_i >= 0, i = 1, ..., m$$

We first normalize the features by scaling the values of each feature to [-1, 1] using the same scaling factor for both the training and test data. This is necessary to prevent features with greater absolute numeric values to dominate those with smaller numeric values. Also, since the kernel values typically involve the inner products of feature vectors, normalizing the values prevents numeric problems such as overflows

The performance of an SVM model depends on the kernel used, the parameters of the kernel, and the soft margin parameter C. We will attempt to optimize each of these.

A. Selection of Kernel

I investigate some commonly used kernels (linear, polynomial, Gaussian radial basis function, and sigmoid) and compare performance. I used LibSVM with default settings (C-SVC, C = 1, γ = 1/# of features, d = 3), and trained the model with the first 70% of the loans and tested the models on the last 30% of the loans in our dataset.

$$Linear : K(x, z) = x^T z$$

$$Polynomial : K(x, z) = (\gamma(x^T z + 1))^d$$

$$RBF : K(x, z) = \exp(-\gamma||x - z||^2)$$

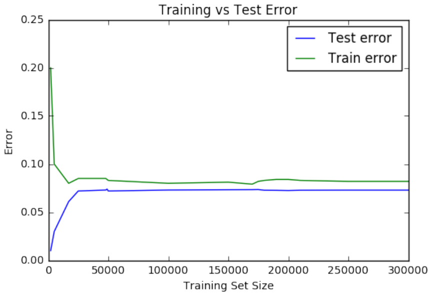
$$Sigmoid : K(x, z) = tanh(\gamma x^T z + d)$$

TABLE II. PERFORMANCE OF SVM WITH VARIOUS KERNELS

Kernel	Accu	Prec	Sens	Spec	G-mean
Linear	92.0	95.6	95.3	79	86.7
Polynomial	92.0	93.5	96.6	60.8	76.6
RBF	91.8	92.9	97.0	56.5	74.03
Sigmoid	89.8	90.8	97.2	40.0	62.3

B. Bias vs. variance

I ran SVM with a linear kernel with variable number of training examples, then compared the training error with the test error to determine whether we are likely to be encountering high bias or high variance in our SVM model with the dataset that we have.



Training vs. Test specificity

The test and training errors converge quickly relative to the number of training examples available, and the gap between them is small, suggesting a high bias in the model. Thus we will increase the number of features. Adding the words selected via TF-IDF as boolean features, we see an increase across all performance metrics, including a 0.7% boost in G-mean.

TABLE III. TABLE TO TEST CAPTIONS AND LABELS

C	Accu	Prec	Sens	Spec	G-mean
10^{-2}	91.7	92.8	97.1	55.8	73.6
1	92.7	95.9	95.3	79.0	86.7
10^2	92.0	95.7	94.3	79.0	86.3

TABLE IV. PERFORMANCE OF SVM WITH VARIOUS OPTIMIZATIONS

Step	Accu	Prec	Sens	Spec	G-mean
Linear Kernel	92.0	95.6	94.3	79.0	86.3
Add features	92.7	95.9	95.3	79.5	87.0
Tune C	92.7	95.9	95.3	79.5	87.0

C. Soft margin parameter

I experimented with different values of the soft margin parameter C. I ran linear kernel SVM with C = $\{10^{-2}, 1, 10^2\}$. Ultimately, I found that using C = 1 yielded the best performance.

VII. RESULTS

From the comparison of each model, I found that SVM has the highest specificity (79%), and thus predicting expired contracts is best with SVM.

VIII. CONCLUSION

From the comparison of multiple models, including Logistic Regression, and SVM and different fine-tuning mechanisms, I found that SVM performs the best(slightly) at predicting expired contracts (optimizing specificity). If we apply the best performing model and prioritize the set of contracts which has lower likelihood of renewals with the right incentives identified from the model like bundled offers, appropriate discount, dedicated account managers among others, we can increase the renewal rate by 47%. Also, majority of the contracts(70%) were predicted from the model 90 days before the actual expiry providing an opportunity window to the Sales team to pursuade the customers with right incentives. Some future work that could further improve the prediction includes:

1.Build Naive Bayes model with various distributions (Gaussion, Bernoulli) to understand if it's performance is better than SVM or Logistic

2.Include in additional features like net promoter score to capture customers sentiments, features that capture industry level trend because whether customer's industry is expanding or contracting would be a good indicator for renewal of contracts

REFERENCES

[1] M. Kubat and S. Matwin, Addressing the curse of imbalanced training sets: One-sided selection," in Proceedings of the Fourteenth International Conference on Machine Learning Morgan Kaufmann, 1997.

[2] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, A Practical Guide to Support Vector Classification, 2010 Available at: https://www.csie.ntu.edu.tw/ cjlin/papers/guide /guide.pdf.

[3] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:127:27, 2011. Software available at: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[4] He, He, and Ali Ghodsi. ”Rare class classification by support vector machine.” Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010

[5] Salazar, D.A. et al. 2012. Comparison between SVM and Logistic Regression: Which One is Better To Discriminate? Revista Colombiana de Estadstica. 35, 2 (2012), 223-237

[6] Chen, Yi-Wei, and Chih-Jen Lin. ”Combining SVMs with Various Feature Selection Strategies.” Feature Extraction Studies in Fuzziness and Soft Computing (2006): 315–24. Web

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