TA2_Models_Bayesian_Classification

Introduction

The objective is to build a Bayesian model to predict successful sales opportunities before we assign engineering resources. We have data from the proposal management system that tracks RFPs *(Request for Proposals)* recieved for engineered products. After intial transformation, the following dimensions are included in the analysis:

- SampleID
- RSF *(Relationship Strength Factor)*
- QuoteDiff *(diffence between our quote and primary competitor quote)*
- RFPDiff *(difference between the dates the RFP response was requested, and when it was returned)*
- ATPDifference *(diffeence between the available to promise - ATP - date and the date required)*
- Result *(whether the opportunity was won or lost)*

Model

The data are not hierarchical - each observation is exchangable, so a single-level, multiple regression model is used, which is transformed to a classification model using a logit function.

A few final transformations were implemented:

- The ATPDifference as scaled down *(divided by 1000)* to bring it into scale with the other dimensions and help the sampler.
- Data was divided into using a holdout validation set of 100 observations, with the rest for training.
- RSF is an ordinal factor, so it is transformed directly to an integer which will easily work within a logistic regression equation.

Model Development and Training

Model Testing

The approach to testing is to pull the estimated parameters out of the sampler, analyze the distributions of the parmaeters and evaluate test data using those parameters in a test set.

To do this, we use a typical logistic regression equation format:

$P(y) = exp(b_0 + (b_1X...)/exp(1 + exp(b_0 + b_1X...)$

Which converts to the following in R for our parameter set:

$Prob < -(exp(alpha[1] + (beta[1] * test[2] + beta[2] * test[3] + beta[3] * test[4] + beta[4] * test[5])))/(1 + (exp(alpha[1] + (beta[1] * test[2] + beta[2] * test[3] + beta[3] * test[4] + beta[4] * test[5]))))$

This produced a probablity of success, which is also coverted to a bernoulli as described below.
Results

Results were tested with the holdout data. After pulling the parameters from the sampler, we used the above equation to compute probability and then generated a binomial result using the following:

test <- test %>% mutate(Pred = ifelse(Prob < .5, 0, 1))

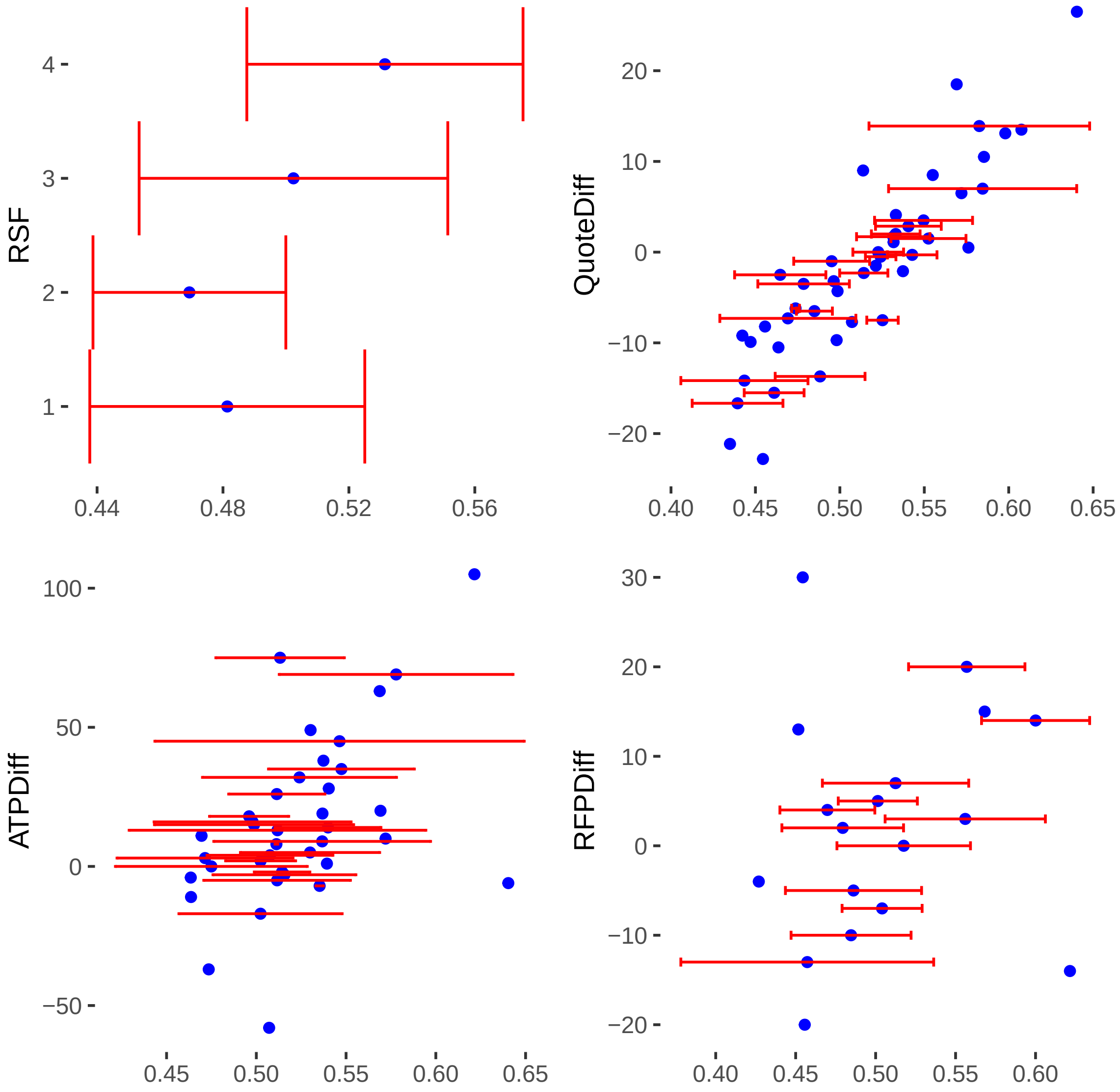
These results were run through a confusion matrix, with metrics as follows:

Confusion Matrix:

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 31  4
##          1 17 48
## 
##                Accuracy : 0.79
##                  95% CI : (0.6971, 0.8651)
##     No Information Rate : 0.52
##     P-Value [Acc > NIR] : 2.098e-08
## 
##                   Kappa : 0.5749
## 
##  Mcnemar's Test P-Value : 0.008829
## 
##             Sensitivity : 0.6458
##             Specificity : 0.9231
##          Pos Pred Value : 0.8857
##          Neg Pred Value : 0.7385
##              Prevalence : 0.4800
##          Detection Rate : 0.3100
##    Detection Prevalence : 0.3500
##       Balanced Accuracy : 0.7845
## 
##        'Positive' Class : 0
## 
```

These results are acceptable *(80% accuracy in sales opportunitis is good - trust me)*.

Results were summarized by parameter and then plotted, comparing the average proability at a 95% confidence interval for each value of the dimension, using code as follows:



Analysis

This all looks good *(consistent with experience)*. The only parameter that raised attention was RSF *(Relationship Strength Factor)* This is a composite index from multiple data sources *(historical W/L, years on account, call frequency, ...)* organized as an ordinal factor *(1 - none, 2 - developing, 3 - good, 4 - strong)*. So, it appeared strange that the probability of success is higher for 1 than a 2. Maybe the models weighting of RSF should be increased? That will be our "hypothesis".

The priors were adjusted to increase the effect of RSF.

This increases the RSF parameter weight, and also tightens the variance *(which expresses an increased confidence)*. The following is the result:

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  0  1
##          0 46 38
##          1  2 14
##
##                Accuracy : 0.6
##                  95% CI : (0.4972, 0.6967)
##     No Information Rate : 0.52
##     P-Value [Acc > NIR] : 0.06623
##
##                   Kappa : 0.2212
##
##  Mcnemar's Test P-Value : 3.13e-08
##
##             Sensitivity : 0.9583
##             Specificity : 0.2692
##          Pos Pred Value : 0.5476
##          Neg Pred Value : 0.8750
##              Prevalence : 0.4800
##          Detection Rate : 0.4600
##    Detection Prevalence : 0.8400
##       Balanced Accuracy : 0.6138
##
##        'Positive' Class : 0
##
```

Reviewing the matrix, increasing the effect of RSF reduced accuracy on the test set. So the data does not support our "hypothesis".

Could it be that "None" describes a new relationship, and that the sales staff tends to overservice these opporutnities to get them on board? It turned out that this is the reason. So the RSF index was reevaluated to weight frequency of calls differently.

Closing Thoughts

Bayesian modeling increases our ability to analyze complex datasets by providing:

- **Increased Interpretability**. Notice how we were able to analyze each parameter and test the effect of changes. This provides a basis for understanding and testing specific effects *(not possible with non-parametric analysis)*
- **Testing of Alternative Hypotheses using Priors**. Priors are used to compromise a model based on data with some blend of experience and prior data. In this example, we didn't use priors to change the model - we used priors to reject a casual hypothesis.
- **Analysis Agility**. Bayesian models adapt more easily to new data and new questions as demonstrated here.