Cogitativo: Claims Denial Prediction

Goal: Predict the claims that fall into denial.code "F13", "J8G", "JO5", "JB8", "JE1", "JC9", "JF1", "JF9", "JG1", "JPA", "JES"

Final Results: Random Forest, 95.8% Sensitivity / 99.6% Specificity on 10% Testing data

Summary

Tools: R (dplyr, data.table, randomForest, caret, xgboost)

1. **Data Cleaning**
    a. Removing predictors which may not be appropriate for this exercise
        i. Claim.Number
            - It is a counting index which should not have any predictive power on future claims.
        ii. Member.ID
            - Although it maybe a good indicator for future denial, for the training and testing split and evaluation process, it should not be included.
            - “Blacklisted” indicator can be considered on the prediction on unseen data only.
        iii. Claim.Line.Number
            - Similar to Claim.Number, it is just a counting index and should be used as a predictor even if it has any predicting power in the training dataset.
            - One assumption embedded here is that **each claim line is independent for each Claim.Number / Member.ID**. This assumption may not be true but it is more convenient for the testing and training evaluation process.

    b. Re-code some variables for convenience of modeling
        i. Create an indicator variable “target” if the claims are denied under the specified denial.code.
        ii. To support randomForest modeling in R, the following predictors are split into multiple columns to control the number of levels
            - Revenue.Code => Revenue.Code1,2,3
            - Service.Code => Service.Code1,2,3
            - Procedure.Code => Procedure.Code1,2,3,4,5
            - Diagnosis.Code => Diagnosis.Code1,2,3,4,5

    c. Training and Testing split
        i. 90% training and 10% testing split
            - The dataset is randomly divided into training and testing set
            - The denial ratio is checked for consistency between training and testing

2. Modeling

a. Algorithm (Random Forest)
   i. Since it is a classification problem with multiple denial codes considered, tree methods should be a good start.
   ii. Random Forest is fast to tune, run and test.
   iii. Other methods considered: boosted trees – No OOB error improvements over RF over a reasonable amount of time on tuning.

b. Trade-off between sensitivity and specificity (Down-sampling)
   i. This dataset is heavily unbalanced (400:1). To have a reasonable sensitivity, re-sampling method must be considered during the modeling building process.
   ii. 5:1 down-sampling ratio is used in the final model to strike for a reasonable balance between FP and FN.

3. Results

a. The final model is a Random Forest with mtry = 7, 8500:1700 down-sampling for each tree.
b. Prediction results on the test set

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 46877     8
         1   187   184

               Accuracy : 0.9959
                 95% CI : (0.9953, 0.9964)
    No Information Rate : 0.9959
    P-Value [Acc > NIR] : 0.6042

                  Kappa : 0.6518
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.958333
            Specificity : 0.996027
         Pos Pred Value : 0.495957
         Neg Pred Value : 0.999829
             Prevalence : 0.004063
         Detection Rate : 0.003894
   Detection Prevalence : 0.007851
      Balanced Accuracy : 0.977180

       'Positive' Class : 1
```

Appendix: Important Predictors