ISQA 8080 Project - Milestone 2
12/10/52019
Eric Troudt
Jonathan Prosser

Project Description:

Who is West?

West Corporation (currently being re-branded as Intrado) provides technical solutions within various business contexts in order to optimize communications between clients, customers, and other companies. One of its main divisions *Unified Communications,* focuses on the facilitation of various business communication mediums in a technologically-enhanced and highly efficient manner, including voice conferencing, IT networks, webinars, etc.

What does the company want?

As with any business, West constantly searches for ways to increase client retention, searching for potential factors that can influence a particular client's likelihood to 'churn' or no longer use West for its business needs. In order to identify those clients with a high risk of churning, West wants to utilize statistical learning methods to generate predictive models able to robustly determine which clients fall into this category.

What is our project?

As part of our project goal, we will seek to provide West's Client Engagement team with accurate predictions of churn probability for any given group of clients. Our project will therefore consist of cleaning and processing data that West considers relevant to client churning, followed by the selection and implementation of a classification-based statistical learning approach that can effectively inform West about which clients it should allocate more time and resources towards to minimize the company's churn rate.

Data Understanding

Our data as a whole comes in the form of 3 CSV files and 1 Excel file. The first CSV file, labeled "firmographic data," is the raw data consisting of 41 variables and just over 55,000 observations. Although useful as a reference, we won't be focusing on this dataset.

The other two CSV files are our "model dataset," consisting of 24,649 observations, and our "test dataset," consisting of 8,184 observations. Both the test and model datasets have the same 7 explanatory variables, these being company number, company creation date, total products, total transactions, total revenue, total usage, and total accounts. The company number is not a true explanatory variable and will serve more as a reference number than anything else.

Our last file, the Excel file, is a data dictionary giving an overview of the meaning behind each variable. They are as follows.

Company Number: A unique identifier indicating which company's information we are looking at - can be as few as 2 digits or as many as 6 digits. Will be used as a reference and not an explanatory variable.

Company Creation Date: Tells the day, month, year, and time that a company's information was first loaded into the system. At this point, we're thinking this variable will not be useful for our model. Imagine two companies that both first opened accounts in April of 2001. One company still has an open account, the other churned October of 2001. Here we have two inherently different scenarios, and yet the same creation date for both companies. I don't see any way a statistical model could capture this accurately. Or alternatively, it could very well be the case that as the creation date gets older, there is a higher chance of the company churning, but this would only make sense. As time passes, there are more opportunities for something to go wrong. But even if this is the case, knowing this doesn't help the business scenario. You can't do anything to stop the passage of time and keep companies from churning. Our preliminary analysis seems to indicate the first scenario. We plan on exploring this variable further regardless. We've managed to split the variable into 5 classes to make it more manageable.

Total Products: A quantitative variable indicating the number of products and services available from a company. Skewed towards 1, with mean of 1.587 and max of 10.

Total Transactions: A quantitative variable indicating the number of transactions or billings made with a company. Range of 1 to 1,491 with mean of 46.24.

Total Revenue: A quantitative variable indicating the total value of charges incurred by a company. Widest range with minimum of -781,981 and maximum of 20,015,125. Mean is 33,389.

Total Usage: A quantitative variable indicating a company's total usage in minutes. Has wide range with minimum of 0 and maximum of 861,482,918. Mean is 667,517.

Total Accounts: A quantitative variable indicating the total number of accounts that a company has opened. Skewed towards 1, with mean of 2.487 and max of 2,618.

Churned: Our target response variable, qualitative, with 0 indicating the company has not churned and 1 indicating the company has churned. Highly skewed towards 0, with mean of 0.0701.

Data Preparation

Processing Individual Variables:

- Company_Number – Company_Number was dropped right away, since unique identifiers cannot provide any meaningful information to any model.

- Company_Creation_Date was dropped from the dataset due to problems with various models handling variables coded in a date-specific format.
- All categorical variables used in the generation of models were first converted from their numerically encoded values into the labels they represented based off the data dictionary provided for each of the datasets. Each variable was then subsequently reset as a factor with NA values added as additional levels.

Incorporating additional firmographic variables:

- In order to effectively combine variables from the firmographic data, both datasets (training and test) were merged with the firmographic dataset by Company_Number, resulting in 20,140 observations from the training, and 6,645 observations from the test dataset that intersected with the firmographic data.
- For each of the additional variables in the merged dataset, the descriptive statistics provided by the summary function were manually examined to determine which variables could potentially add predictive power to the different classification models. Out of the 42 additional variables provided by the firmographic data, 10 were selected to be used with the current classification models, and were chosen primarily on two separate criteria, namely those variables that had minimal redundancy with existing variables, and also contained relatively few NA values (less than ~ 1,600).

  The 10 variables included:

  - BEMFAB__Marketability_
  - Employee_Count_Total
  - Major_Industry_Category_Name
  - Import_Export_Agent_Code
  - Number_of_Family_Members
  - Location_Type
  - Public_Private_Indicator
  - Legal_Status_Code
  - Population_Code
  - Manufacturing_Indicator

- *Imputing missing values* – The “bagImpute” method from caret’s preProcess function was utilized for imputing the NA values present with all numerical variables.

Determining collinearity (using findCorrelation from caret):

- Using a cutoff threshold of 0.65, caret’s findCorrelation was used to analyze the correlation matrix of all numeric variables in the model, with both total_transactions and total_revenue exhibiting ~0.75. The robustness to collinearity of both decision trees and support vector machines (the final models selected for the project) allow for all of these variables to remain.

Centering/Scaling

- To address the pronounced differences in scales between the numerical predictor variables, the center and scale options were passed to the preProcess function, effectively transforming the numerical variable data into normal distributions with mean 0.

Applying SMOTE for oversampling:

- To account for the strong disparity between not-churned (93%) and churned (7%) values within the training data, smote was specified in the trainControl function for applying an oversampling approach to inflate the churn data with artificial values. Smote was also applied outside of caret in order to apply different values for k (k-nearest neighbors) to determine if this parameter impacted the models.

Resampling

- Training data was partitioned into a 70:30 split (training:validation) to account for the missing response variable in the test dataset provided to us.
- Cross-validation with 10 folds was used for estimating the performance of each model.

Model Implementations

- model types:
    - Logistic Regression
    - LDA
    - QDA
    - KNN
    - Random Forest Decision Tree
    - Gradient Boosted Decision Tree
    - Support Vector Machines (linear, poly, and radial)

- model forms:
    - All model forms consisted of churned as the dependent variable and various combinations of the predictors variables previously described. Each model form omitted Company_Number which only serves as an identifier and does not provide any meaningful information.

DATA MODELING, MODEL BUILDING

Model Training/Evaluation

For the purpose of evaluating our models, we focus first on the receiver operating characteristic (ROC) curve. ROC (or AUC for area under the curve) is a good measure of overall performance, taking into account both sensitivity and specificity at all thresholds. Having a high ROC generally indicates good overall strength. ROC was the metric used in cross-validation to determine the optimal value for our tuning parameters.

Although ROC was our primary metric of evaluation, ROC alone cannot identify a good model. In this case, our response variable was highly skewed towards a negative response (not churned). A model that simply predicts “No” in all cases could easily achieve over a 90% success rate, but this has no value in practice. In order to build a useful model, we will also need to achieve a high sensitivity score. Sensitivity is a measure of how well our model is able to identify positive cases, in this context meaning our ability to accurately identify companies that do churn. It is our business goal to be able to identify these companies before they churn, and we cannot accomplish this without some level of predictive sensitivity in our model.

We also made sure that specificity scores fell within a reasonable range, although this wasn’t an issue with most models as specificity scores were generally high.

For this reason, we are looking primarily at both ROC and sensitivity as we evaluate our models. Although a high ROC is important for understanding the overall quality of a particular model, it is reasonable to sacrifice a small reduction of ROC in order to greatly improve our sensitivity. Exactly what trade off is optimal, however, is difficult to determine, as it depends on the real-world costs associated with our particular business situation. We can only guess based on the information that’s given.

We considered all classification models available to us as potentially viable. This includes logarithmic regression, linear discriminant analysis, quadratic discriminant analysis, KNN, pruned classification trees, random forests, gradient boosted trees, and support vector machines with linear, polynomial, and radial kernels. Ultimately, it is fairly easy to run different model types in R, and we felt like leaving any model out would be a failure on our part.

That being said, we were able to eliminate certain models very quickly. Based on initial run throughs, random forest and gradient boosted trees stood out as strong performers. These two models typically had AUC’s several points higher than other model types. Although they didn’t initially score as high on sensitivity, we found that by shifting the threshold from 0.5 to 0.45 or even 0.4, we could increase our sensitivity at the cost of some specificity while still outperforming other model types overall.

Our best model was the gradient boosted tree. Once settled on this model type as a potential final form, we then used the variable importance function to systematically eliminate variables that were contributing the least to our model. After some trial and error, we eventually settled on a version that uses 8 explanatory variables, these being Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members. We tried removing Business_Code, but the drop in AUC was more than we were willing to accept.

After noticing that one of our other models performed better using these 8 explanatory variables, we then went back through the other model types to check for improvements. Although most models performed better using these 8 variables, none out-performed our gradient boosted model.

Scores on test data for the gradient boosted model with 8 variables were accuracy=0.8068, sensitivity=0.4771, specificity=0.8353, and AUC=0.730. Through cross-validation, the final parameter values chosen for the model were nrounds = 100, max_depth = 2, eta = 0.4, gamma = 0, colsample_bytree =0.8, min_child_weight = 1 and subsample = 1.

Parameter Tuning and Selection

For all models with tuning parameters, the tuning of each parameter was performed using K-fold cross-validation with K=10. Prefered parameter values were chosen based on highest ROC score, unless noted otherwise.

For KNN, K=150 was chosen for its high ROC and sensitivity values.

For Both Decision trees and support vector machines, numerous parameters were hypertuned in order to select the optimal parameter value for a given model.

Random Forest decision tree: The ‘mtry’ parameter is the only parameter available for tuning in the random forest model, and can be modified to allow different numbers of predictor variables to be randomly selected from at each split. This parameter was set to mtry=3 based on cross-validation results..

Gradient-boosted decision tree: 3 different parameters were tuned for the xbgTree models: nrounds, maxdepth, and eta. The descriptions and values used for each parameter were as follows:

- nrounds: Controls the number of iterations (trees) that are used in generating the model, which is necessary in order to prevent overfitting from the data. Ten different values ranging from 10 - 100 were applied.
- maxdepth: Controls the number of splits made at each iteration during the generation of the model. Four different values were applied (1, 3, 5, 7).
- eta: Controls the rate of ‘learning’ for the model, i.e. the extent to which each tree’s results are incorporated into the model. Three values were applied (0.001, 0.01, 0.1)

Support Vector Machines: Up to 4 different parameters were tuned for the svm models depending on which kernel was utilized: C, degree, sigma, and scale. The descriptions and values used for each parameter were as follows:

- C (linear, poly, radial): Controls the extent to which observations are incorrectly (within margin or on the other side of the decision boundary) classified. Values were selected at random using the ‘random’ option in caret’s trainControl.

- sigma (radial): Controls the flexibility/smoothing of the radial kernel. Smaller values for sigma will more tightly fit the predictors, increasing prediction accuracy, but also the variance of the model.
- degree (poly): Determines the degree of the polynomial used in the kernel. Values were selected at random using the ‘random’ option in caret’s trainControl.
- Scale or gamma (poly): Controls the extent to which the model will capture the complexity of the dataset, i.e. higher gamma values will more readily separate non-linear data. Values were selected at random using the ‘random’ option in caret’s trainControl.

EVALUATION, RECOMMENDATION, AND CONCLUSION

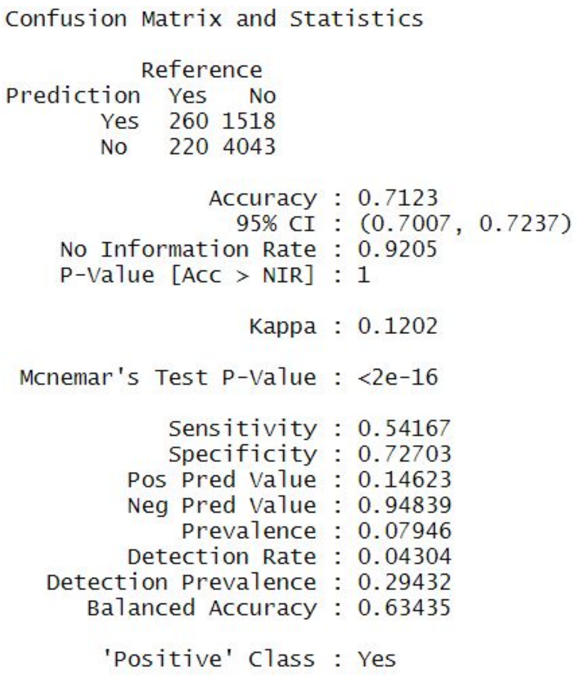
Model Results

Logarithmic Regression

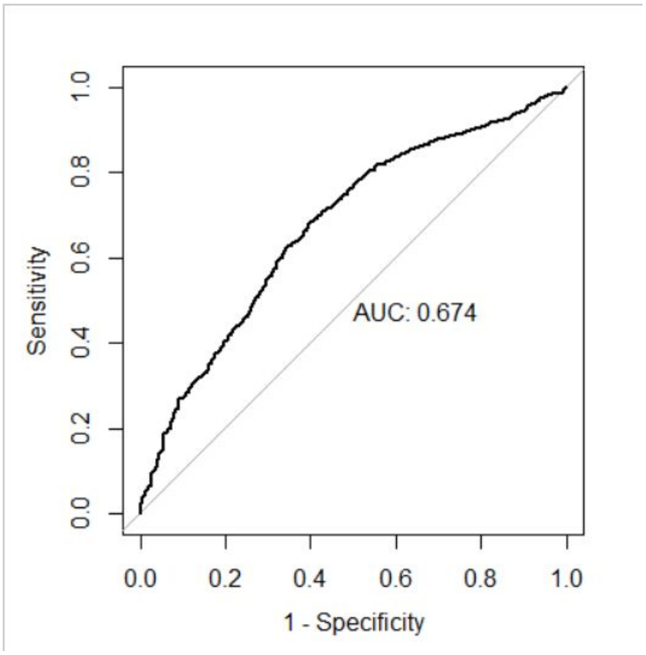
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling.

Test Performance



```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  260 1518
       No   220 4043

               Accuracy : 0.7123
                 95% CI : (0.7007, 0.7237)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1202

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

            Sensitivity : 0.54167
            Specificity : 0.72703
         Pos Pred Value : 0.14623
         Neg Pred Value : 0.94839
             Prevalence : 0.07946
         Detection Rate : 0.04304
   Detection Prevalence : 0.29432
      Balanced Accuracy : 0.63435

       'Positive' Class : Yes
```

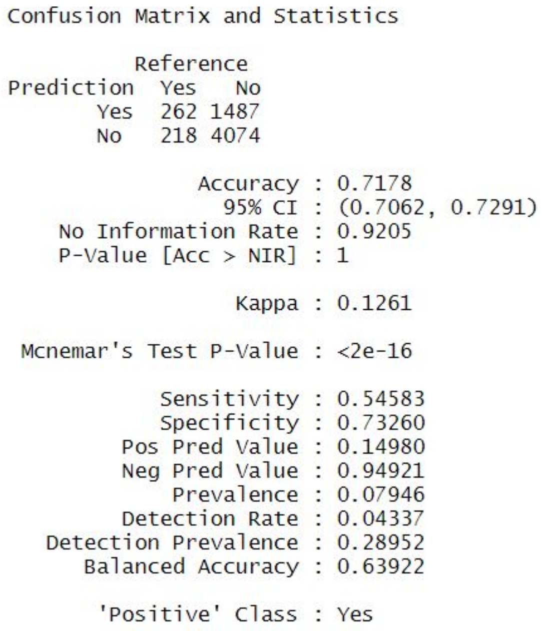


Linear Discriminant Analysis

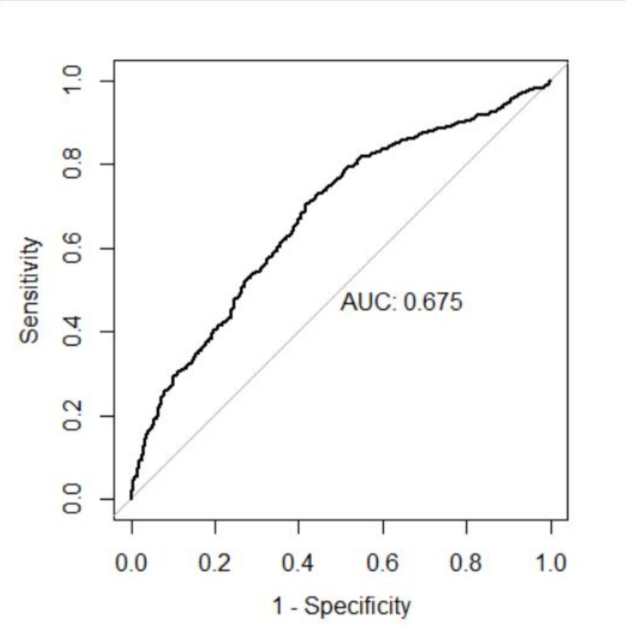
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling.

Test Performance



```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  262 1487
       No   218 4074

               Accuracy : 0.7178
                 95% CI : (0.7062, 0.7291)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1261

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

            Sensitivity : 0.54583
            Specificity : 0.73260
         Pos Pred Value : 0.14980
         Neg Pred Value : 0.94921
             Prevalence : 0.07946
         Detection Rate : 0.04337
   Detection Prevalence : 0.28952
      Balanced Accuracy : 0.63922

       'Positive' Class : Yes
```

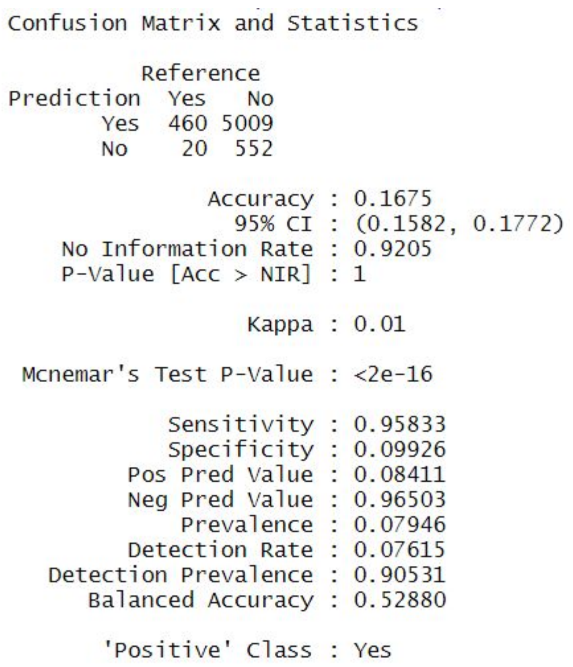


Quadratic Discriminant Analysis

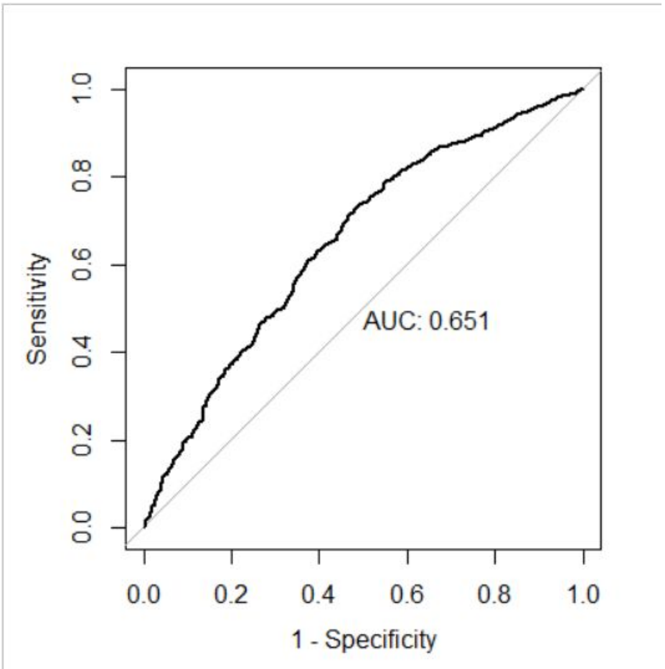
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling.

Test Performance



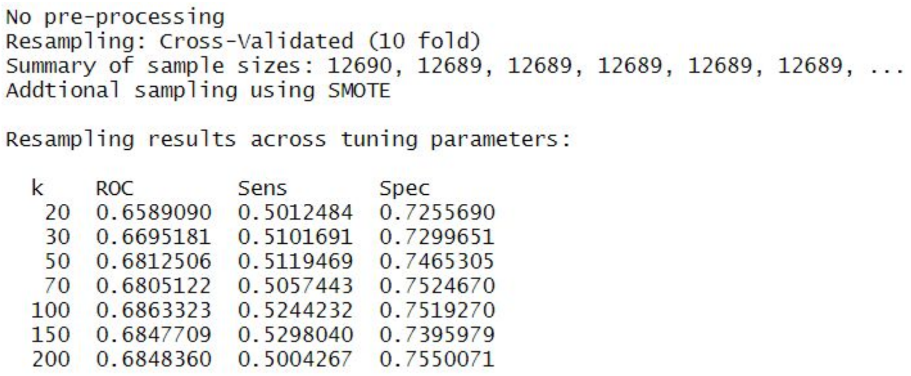
```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  460 5009
       No    20  552

               Accuracy : 0.1675
                 95% CI : (0.1582, 0.1772)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : 1

                  Kappa : 0.01

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

            Sensitivity : 0.95833
            Specificity : 0.09926
         Pos Pred Value : 0.08411
         Neg Pred Value : 0.96503
             Prevalence : 0.07946
         Detection Rate : 0.07615
   Detection Prevalence : 0.90531
      Balanced Accuracy : 0.52880

       'Positive' Class : Yes
```



<u>KNN</u>

Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

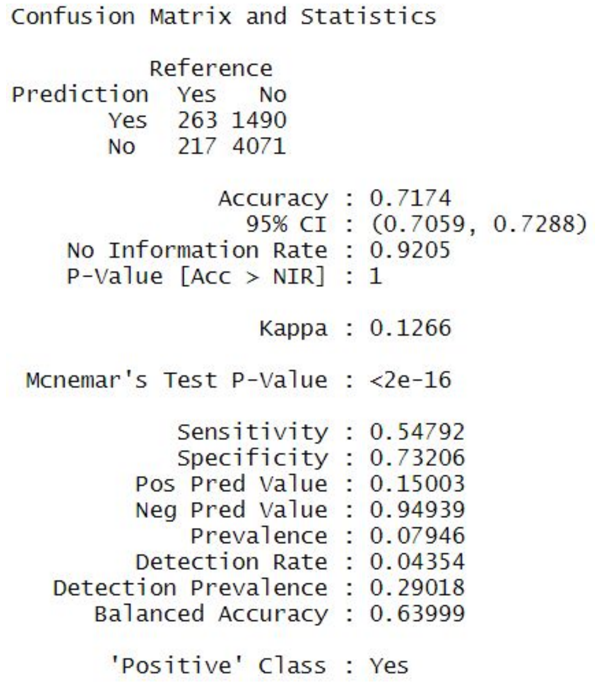
Using SMOTE for sampling.



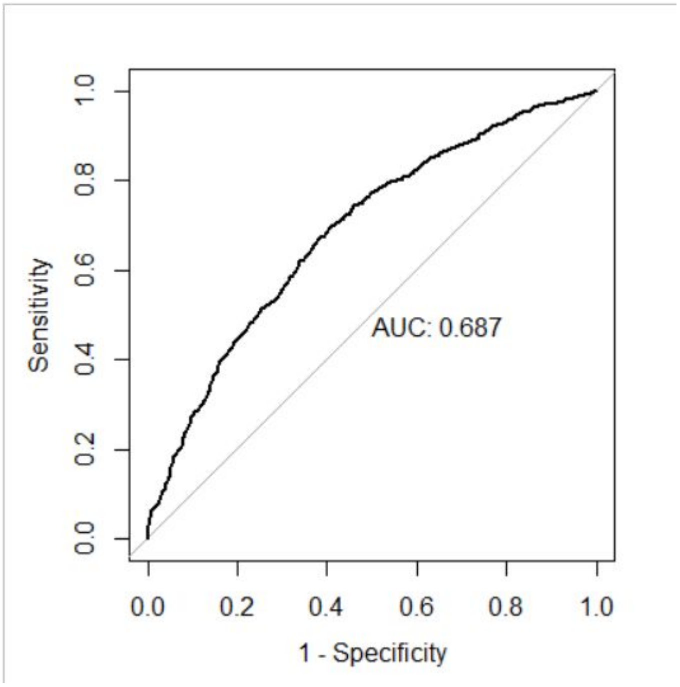
```
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 12690, 12689, 12689, 12689, 12689, 12689, ...
Addtional sampling using SMOTE

Resampling results across tuning parameters:

  k    ROC        Sens       Spec
   20  0.6589090  0.5012484  0.7255690
   30  0.6695181  0.5101691  0.7299651
   50  0.6812506  0.5119469  0.7465305
   70  0.6805122  0.5057443  0.7524670
  100  0.6863323  0.5244232  0.7519270
  150  0.6847709  0.5298040  0.7395979
  200  0.6848360  0.5004267  0.7550071
```

We chose K=150 for high ROC and best sensitivity.

Test Performance



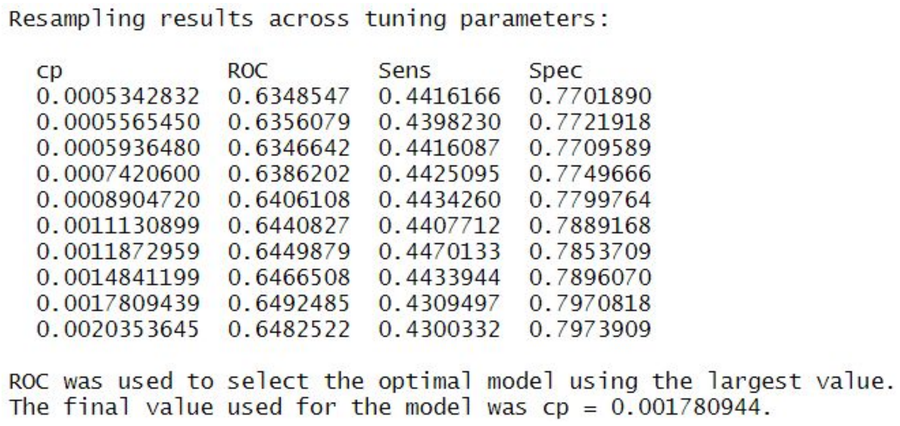
```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  263 1490
       No   217 4071

               Accuracy : 0.7174
                 95% CI : (0.7059, 0.7288)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1266

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

            Sensitivity : 0.54792
            Specificity : 0.73206
         Pos Pred Value : 0.15003
         Neg Pred Value : 0.94939
             Prevalence : 0.07946
         Detection Rate : 0.04354
   Detection Prevalence : 0.29018
      Balanced Accuracy : 0.63999

       'Positive' Class : Yes
```



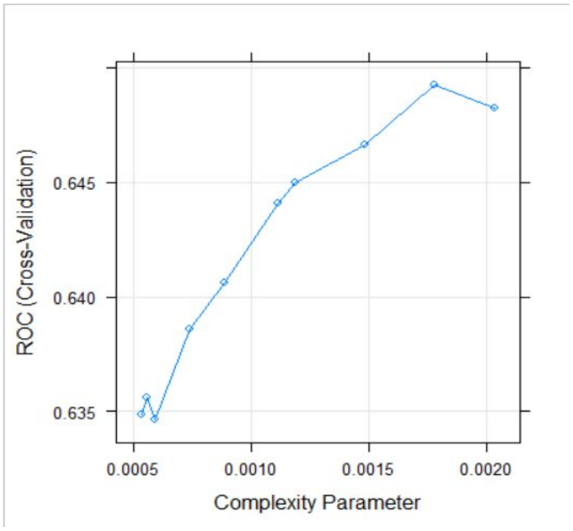
Pruned Classification Tree

Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

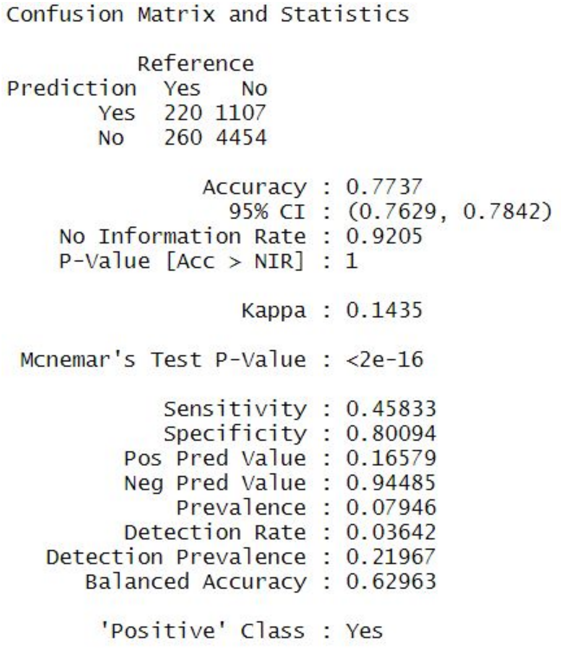
Using SMOTE for sampling.



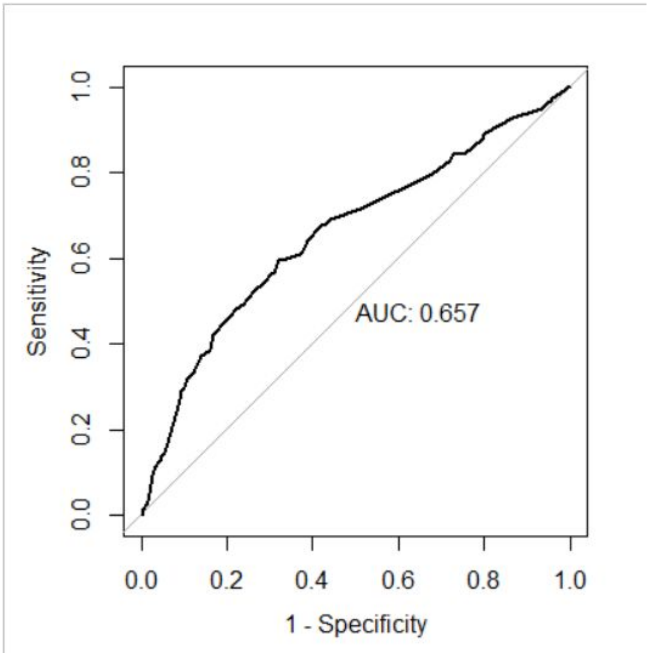
```
Resampling results across tuning parameters:

  cp            ROC        Sens       Spec     
  0.0005342832  0.6348547  0.4416166  0.7701890
  0.0005565450  0.6356079  0.4398230  0.7721918
  0.0005936480  0.6346642  0.4416087  0.7709589
  0.0007420600  0.6386202  0.4425095  0.7749666
  0.0008904720  0.6406108  0.4434260  0.7799764
  0.0011130899  0.6440827  0.4407712  0.7889168
  0.0011872959  0.6449879  0.4470133  0.7853709
  0.0014841199  0.6466508  0.4433944  0.7896070
  0.0017809439  0.6492485  0.4309497  0.7970818
  0.0020353645  0.6482522  0.4300332  0.7973909

ROC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.001780944.
```



Test Performance



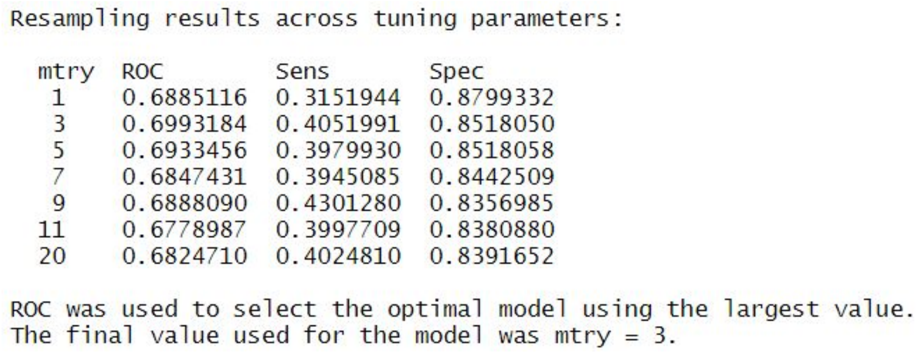
```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  220 1107
       No   260 4454
                                          
               Accuracy : 0.7737          
                 95% CI : (0.7629, 0.7842)
    No Information Rate : 0.9205          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.1435          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 0.45833         
            Specificity : 0.80094         
         Pos Pred Value : 0.16579         
         Neg Pred Value : 0.94485         
             Prevalence : 0.07946         
         Detection Rate : 0.03642         
   Detection Prevalence : 0.21967         
      Balanced Accuracy : 0.62963         
                                          
       'Positive' Class : Yes             
```



Random Forest

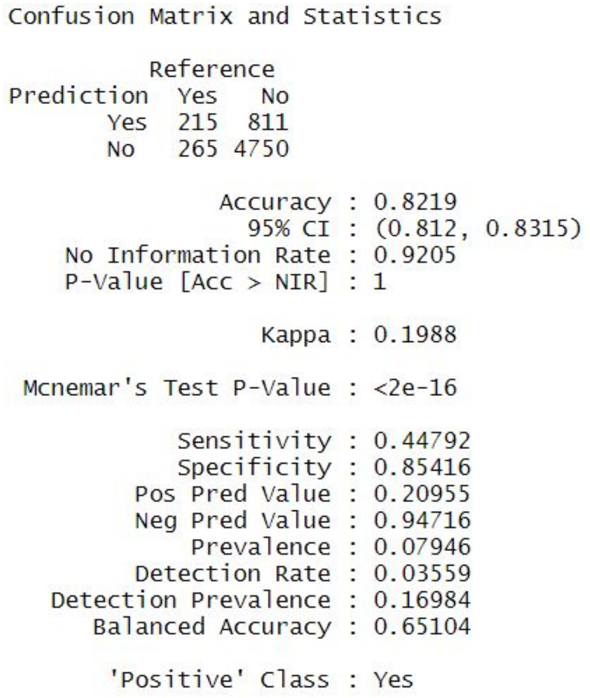
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling.

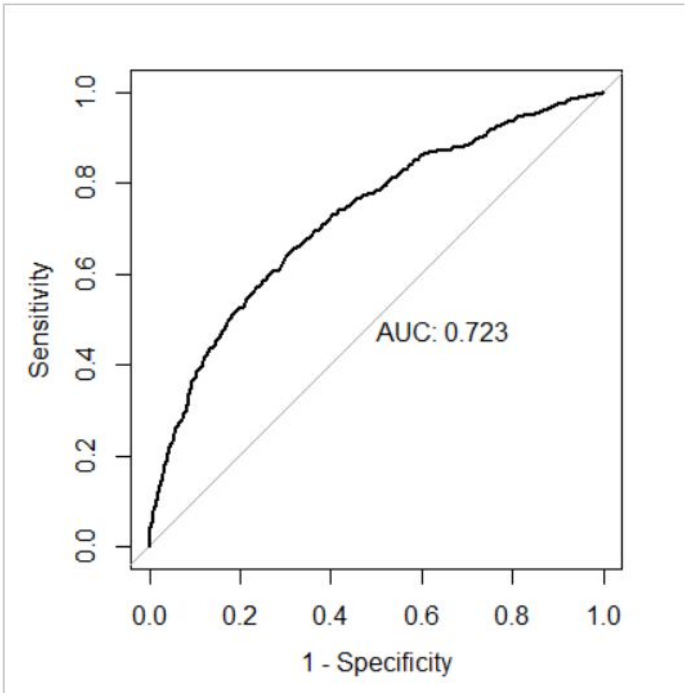


```
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec     
   1    0.6885116  0.3151944  0.8799332
   3    0.6993184  0.4051991  0.8518050
   5    0.6933456  0.3979930  0.8518058
   7    0.6847431  0.3945085  0.8442509
   9    0.6888090  0.4301280  0.8356985
  11    0.6778987  0.3997709  0.8380880
  20    0.6824710  0.4024810  0.8391652

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 3.
```

Test Performance



```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  215  811
       No   265 4750
                                         
               Accuracy : 0.8219         
                 95% CI : (0.812, 0.8315)
    No Information Rate : 0.9205         
    P-Value [Acc > NIR] : 1              
                                         
                  Kappa : 0.1988         
                                         
 Mcnemar's Test P-Value : <2e-16         
                                         
            Sensitivity : 0.44792        
            Specificity : 0.85416        
         Pos Pred Value : 0.20955        
         Neg Pred Value : 0.94716        
             Prevalence : 0.07946        
         Detection Rate : 0.03559        
   Detection Prevalence : 0.16984        
      Balanced Accuracy : 0.65104        
                                         
       'Positive' Class : Yes            
```



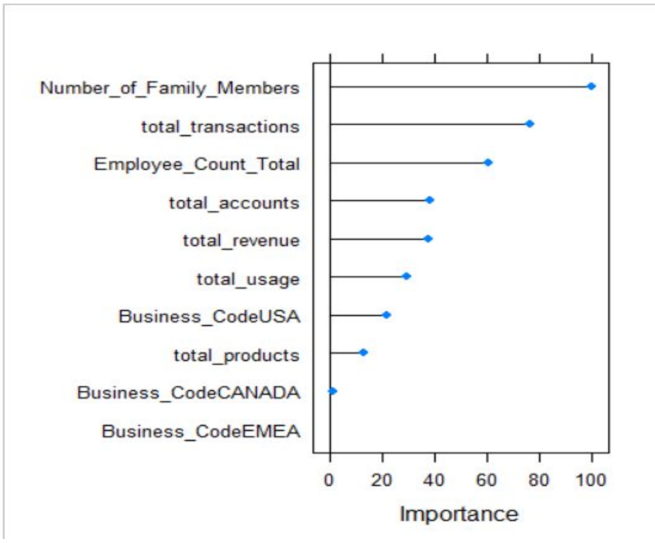
Gradient Boosted Tree

Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

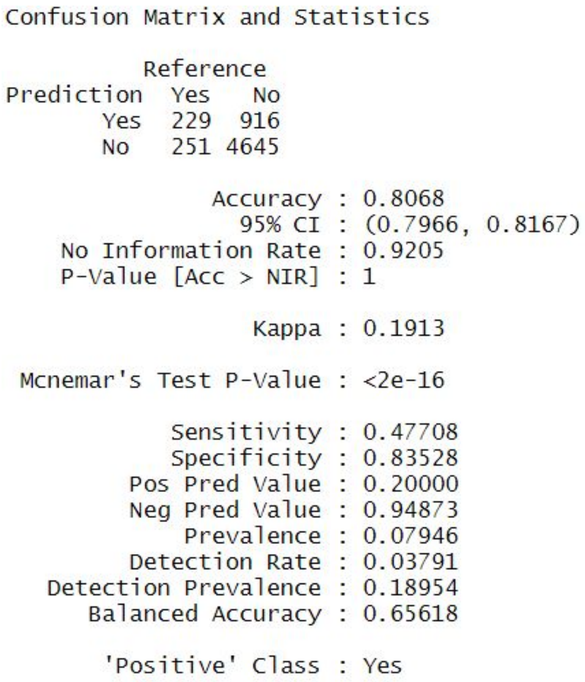
Using SMOTE for sampling.

The final values used for the model were nrounds = 100, max_depth = 2, eta = 0.4, gamma = 0, colsample_bytree =0.8, min_child_weight = 1 and subsample = 1.

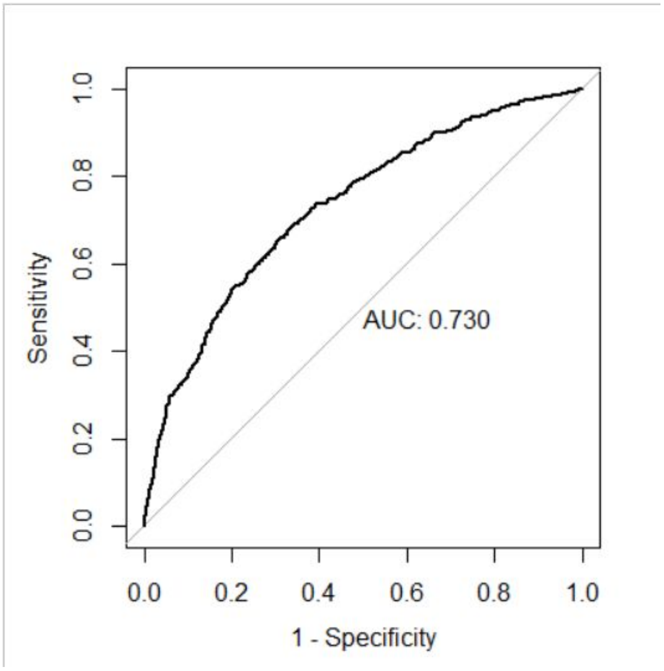
Variable Importance Plot



Test Performance



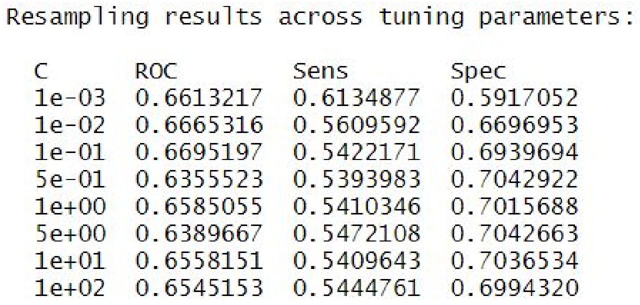
```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  229  916
       No   251 4645

               Accuracy : 0.8068
                 95% CI : (0.7966, 0.8167)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1913

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

            Sensitivity : 0.47708
            Specificity : 0.83528
         Pos Pred Value : 0.20000
         Neg Pred Value : 0.94873
             Prevalence : 0.07946
         Detection Rate : 0.03791
   Detection Prevalence : 0.18954
      Balanced Accuracy : 0.65618

       'Positive' Class : Yes
```



<u>Support Vector Machine - Linear Kernal</u>

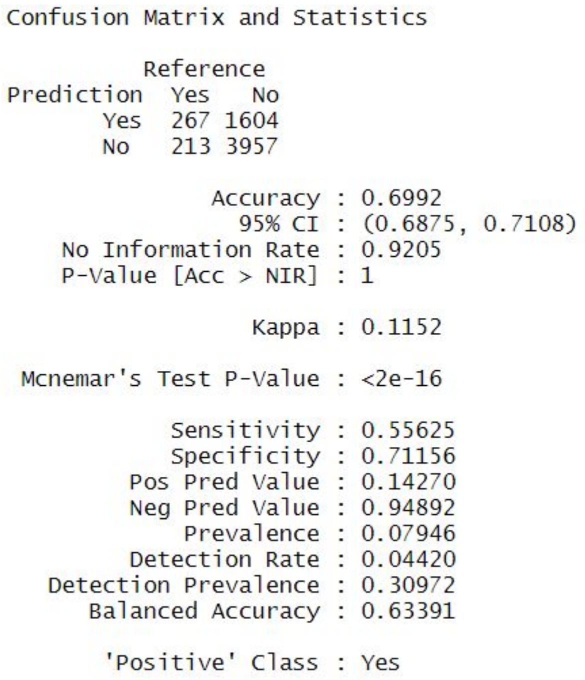
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling and C=0.1

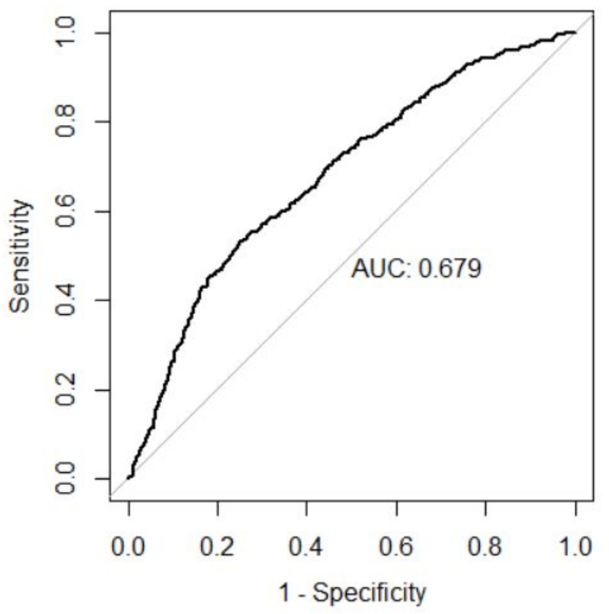


```
Resampling results across tuning parameters:

  C      ROC        Sens       Spec     
  1e-03  0.6613217  0.6134877  0.5917052
  1e-02  0.6665316  0.5609592  0.6696953
  1e-01  0.6695197  0.5422171  0.6939694
  5e-01  0.6355523  0.5393983  0.7042922
  1e+00  0.6585055  0.5410346  0.7015688
  5e+00  0.6389667  0.5472108  0.7042663
  1e+01  0.6558151  0.5409643  0.7036534
  1e+02  0.6545153  0.5444761  0.6994320
```

Test Performance



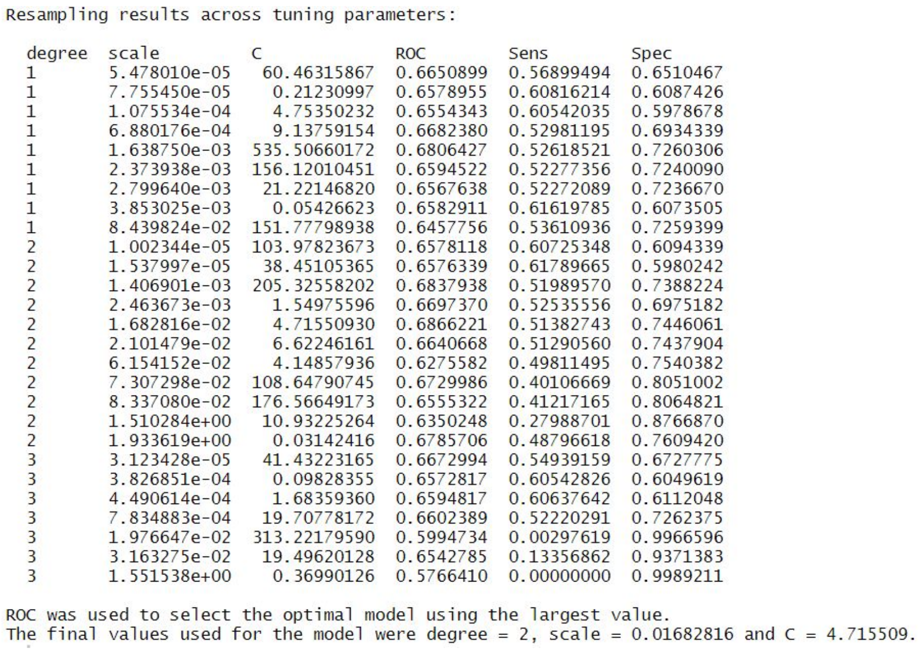
```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  267 1604
       No   213 3957
                                          
               Accuracy : 0.6992          
                 95% CI : (0.6875, 0.7108)
    No Information Rate : 0.9205          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.1152          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 0.55625         
            Specificity : 0.71156         
         Pos Pred Value : 0.14270         
         Neg Pred Value : 0.94892         
             Prevalence : 0.07946         
         Detection Rate : 0.04420         
   Detection Prevalence : 0.30972         
      Balanced Accuracy : 0.63391         
                                          
       'Positive' Class : Yes             
```



Support Vector Machine - Polynomial Kernal

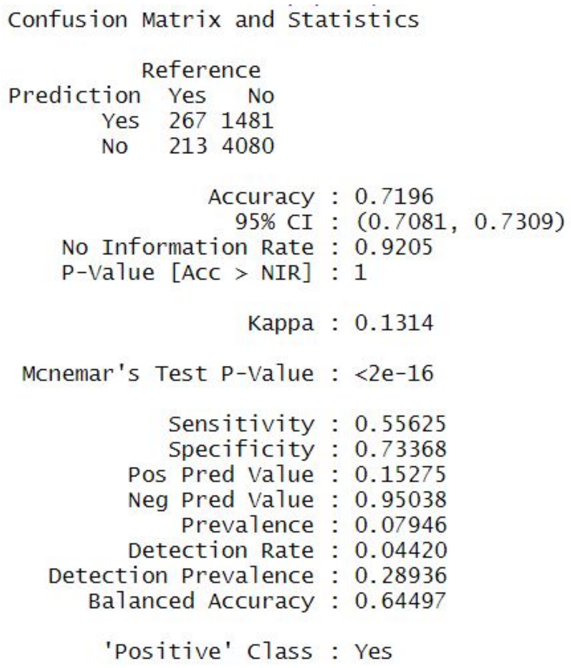
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling, degree=2, scale=0.0168, C=4.716

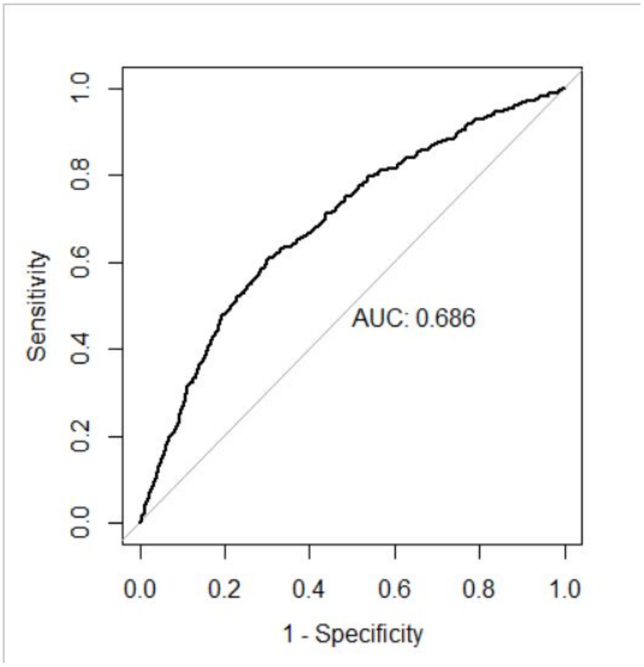


```
Resampling results across tuning parameters:

  degree  scale         C             ROC        Sens        Spec     
  1       5.478010e-05   60.46315867  0.6650899  0.56899494  0.6510467
  1       7.755450e-05    0.21230997  0.6578955  0.60816214  0.6087426
  1       1.075534e-04    4.75350232  0.6554343  0.60542035  0.5978678
  1       6.880176e-04    9.13759154  0.6682380  0.52981195  0.6934339
  1       1.638750e-03  535.50660172  0.6806427  0.52618521  0.7260306
  1       2.373938e-03  156.12010451  0.6594522  0.52277356  0.7240090
  1       2.799640e-03   21.22146820  0.6567638  0.52272089  0.7236670
  1       3.853025e-03    0.05426623  0.6582911  0.61619785  0.6073505
  1       8.439824e-02  151.77798938  0.6457756  0.53610936  0.7259399
  2       1.002344e-05  103.97823673  0.6578118  0.60725348  0.6094339
  2       1.537997e-05   38.45105365  0.6576339  0.61789665  0.5980242
  2       1.406901e-03  205.32558202  0.6837938  0.51989570  0.7388224
  2       2.463673e-03    1.54975596  0.6697370  0.52535556  0.6975182
  2       1.682816e-02    4.71550930  0.6866221  0.51382743  0.7446061
  2       2.101479e-02    6.62246161  0.6640668  0.51290560  0.7437904
  2       6.154152e-02    4.14857936  0.6275582  0.49811495  0.7540382
  2       7.307298e-02  108.64790745  0.6729986  0.40106669  0.8051002
  2       8.337080e-02  176.56649173  0.6555322  0.41217165  0.8064821
  2       1.510284e+00   10.93225264  0.6350248  0.27988701  0.8766870
  2       1.933619e+00    0.03142416  0.6785706  0.48796618  0.7609420
  3       3.123428e-05   41.43223165  0.6672994  0.54939159  0.6727775
  3       3.826851e-04    0.09828355  0.6572817  0.60542826  0.6049619
  3       4.490614e-04    1.68359360  0.6594817  0.60637642  0.6112048
  3       7.834883e-04   19.70778172  0.6602389  0.52220291  0.7262375
  3       1.976647e-02  313.22179590  0.5994734  0.00297619  0.9966596
  3       3.163275e-02   19.49620128  0.6542785  0.13356862  0.9371383
  3       1.551538e+00    0.36990126  0.5766410  0.00000000  0.9989211

ROC was used to select the optimal model using the largest value.
The final values used for the model were degree = 2, scale = 0.01682816 and C = 4.715509.
```

Test Performance



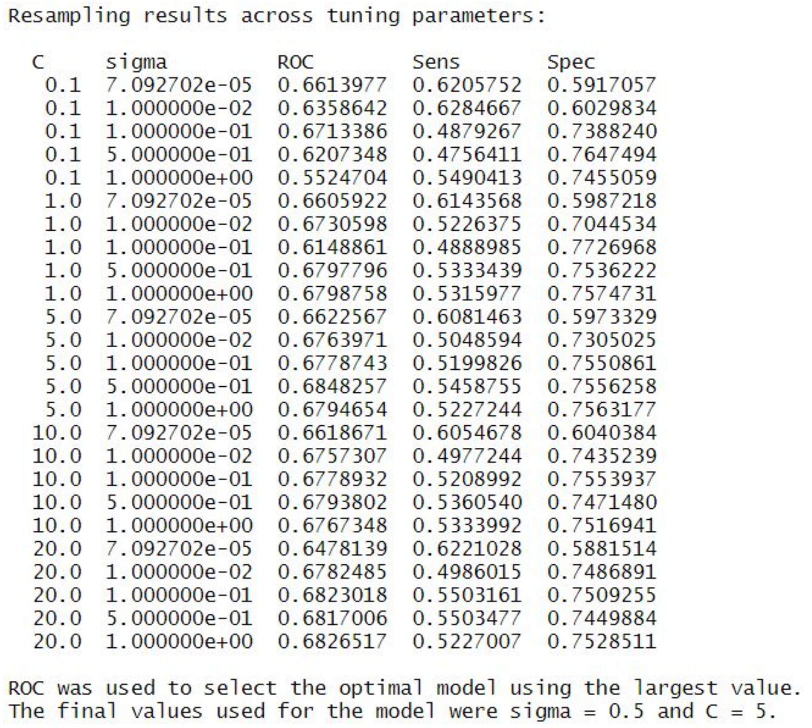
```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  267 1481
       No   213 4080
                                          
               Accuracy : 0.7196          
                 95% CI : (0.7081, 0.7309)
    No Information Rate : 0.9205          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.1314          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 0.55625         
            Specificity : 0.73368         
         Pos Pred Value : 0.15275         
         Neg Pred Value : 0.95038         
             Prevalence : 0.07946         
         Detection Rate : 0.04420         
   Detection Prevalence : 0.28936         
      Balanced Accuracy : 0.64497         
                                          
       'Positive' Class : Yes             
```



Support Vector Machine - Radial Kernal

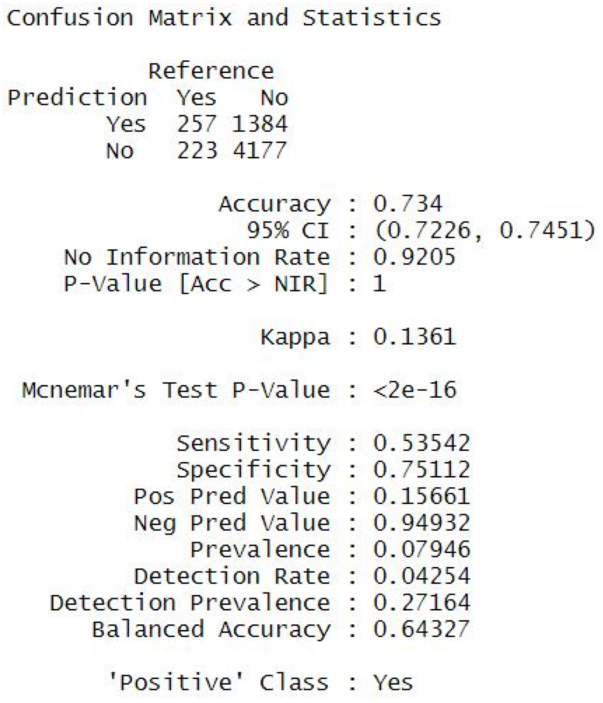
Predictive Variables: Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members

Using SMOTE for sampling.

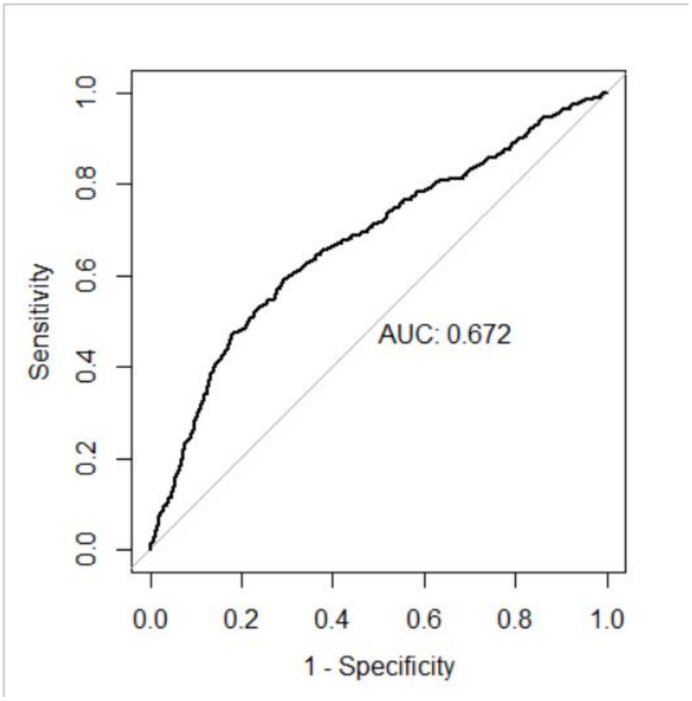


```
Resampling results across tuning parameters:

  C     sigma         ROC        Sens       Spec     
   0.1  7.092702e-05  0.6613977  0.6205752  0.5917057
   0.1  1.000000e-02  0.6358642  0.6284667  0.6029834
   0.1  1.000000e-01  0.6713386  0.4879267  0.7388240
   0.1  5.000000e-01  0.6207348  0.4756411  0.7647494
   0.1  1.000000e+00  0.5524704  0.5490413  0.7455059
   1.0  7.092702e-05  0.6605922  0.6143568  0.5987218
   1.0  1.000000e-02  0.6730598  0.5226375  0.7044534
   1.0  1.000000e-01  0.6148861  0.4888985  0.7726968
   1.0  5.000000e-01  0.6797796  0.5333439  0.7536222
   1.0  1.000000e+00  0.6798758  0.5315977  0.7574731
   5.0  7.092702e-05  0.6622567  0.6081463  0.5973329
   5.0  1.000000e-02  0.6763971  0.5048594  0.7305025
   5.0  1.000000e-01  0.6778743  0.5199826  0.7550861
   5.0  5.000000e-01  0.6848257  0.5458755  0.7556258
   5.0  1.000000e+00  0.6794654  0.5227244  0.7563177
  10.0  7.092702e-05  0.6618671  0.6054678  0.6040384
  10.0  1.000000e-02  0.6757307  0.4977244  0.7435239
  10.0  1.000000e-01  0.6778932  0.5208992  0.7553937
  10.0  5.000000e-01  0.6793802  0.5360540  0.7471480
  10.0  1.000000e+00  0.6767348  0.5333992  0.7516941
  20.0  7.092702e-05  0.6478139  0.6221028  0.5881514
  20.0  1.000000e-02  0.6782485  0.4986015  0.7486891
  20.0  1.000000e-01  0.6823018  0.5503161  0.7509255
  20.0  5.000000e-01  0.6817006  0.5503477  0.7449884
  20.0  1.000000e+00  0.6826517  0.5227007  0.7528511

ROC was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.5 and C = 5.
```

Test Performance



```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  257 1384
       No   223 4177
                                          
               Accuracy : 0.734           
                 95% CI : (0.7226, 0.7451)
    No Information Rate : 0.9205          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.1361          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 0.53542         
            Specificity : 0.75112         
         Pos Pred Value : 0.15661         
         Neg Pred Value : 0.94932         
             Prevalence : 0.07946         
         Detection Rate : 0.04254         
   Detection Prevalence : 0.27164         
      Balanced Accuracy : 0.64327         
                                          
       'Positive' Class : Yes             
```



Evaluation Metrics

Measures used for evaluating model performance included:

- Accuracy - Proportion of observations correctly classified as churned/not churned?
- Sensitivity - Proportion churned observations correctly identified?
- Specificity - Proportion not churned observations correctly identified?
- Area under the curve (ROC) - a Measure of overall performance for the model in terms of correctly identifying both churned/not-churned observations

While the accuracy provided a very general sense of model performance, stronger emphasis was placed on both the sensitivity and specificity of the classifier along with its AUC, which provided a much more robust measurement of the classifier's ability to correctly identify churned observations while minimizing the misclassification of non-churned observations.

Result Interpretations for Client

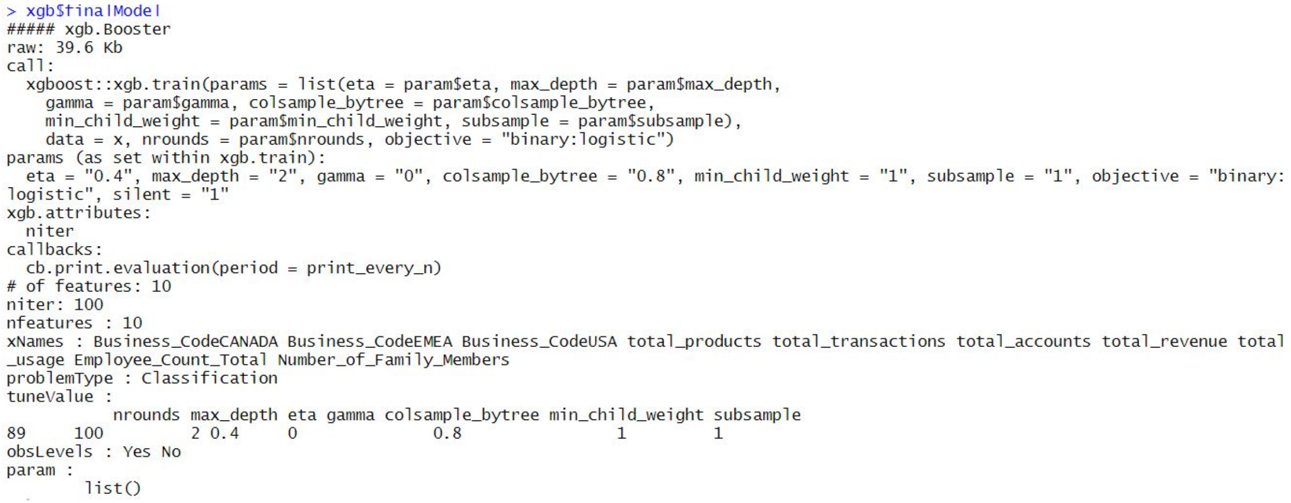
As the results indicate, the sensitivities of both the gradient-boosted and random forest decision trees were the highest out of all models that were tested. The final model used for creating predictions was the gradient boosted decision tree, which had both the most optimal ratio for sensitivity:specificity (47.7:83.5), and also produced the highest AUC (0.73). Indeed there were other models and different combinations of variables that yielded higher specificities at the expense of much lower sensitivities, but we reasoned that these models were less powerful overall. Our rationale for this was as follows:

Assuming that the costs and resources that Intrado/West would allocate towards retaining those customers identified as potential 'churners' by our model are negligible or not significant, then the two models with the highest sensitivities (and manageable specificities) would allow them to maximize their retention rate by giving them the largest pool of potential churners.

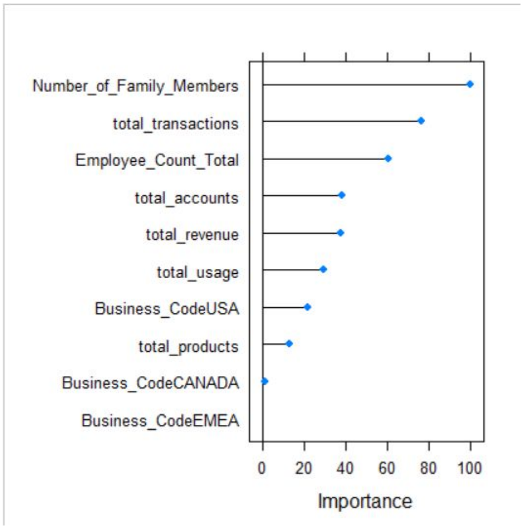
Conclusion

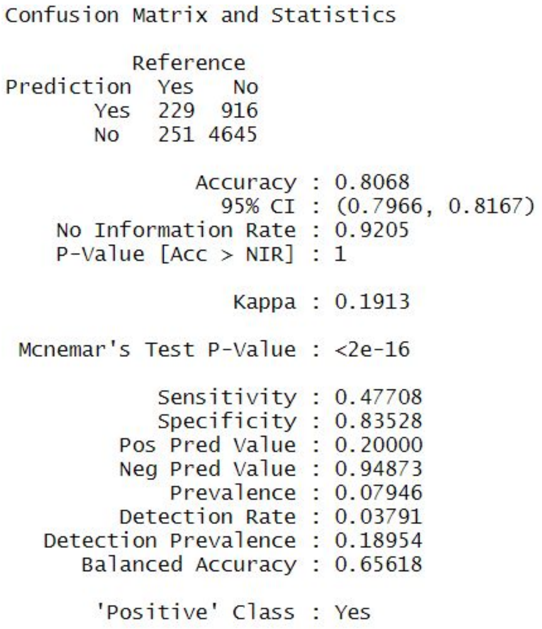
The gradient boosted model with SMOTE sampling, explanatory variables Business_Code, total_products, total_transactions, total_accounts, total_revenue, total_usage, Employee_Count_Total, and Number_of_Family_Members and tuning parameters nrounds = 100, max_depth = 2, eta = 0.4, gamma = 0, colsample_bytree =0.8, min_child_weight = 1 and subsample = 1 has the best predictive power overall. At a base threshold of 0.5 the model performs well, but threshold should be tuned to achieve desired sensitivity/specificity based on costs of business associated with false positives and false negatives. (For example, when the cost of a false positive is low and the cost of a false negative is very high, then higher sensitivity is desirable).

As far as model interpretation goes, the top three variables that contribute the largest relative influence within the gradient boosted tree are Number_of_Family_Members, total_transactions, and Employee_Count_Total, which we find to be a significant result, since two of these variables were not originally present in the datasets provided by West/Intrado, and yet they exert large influences on the prediction accuracy of the top model. We therefore conclude that the scope of a client's corporate structure and the size of its employee base, along with its transaction activity are the most crucial indicators of churning, and further analysis or examination of these variables may provide additional insight into churn probability.

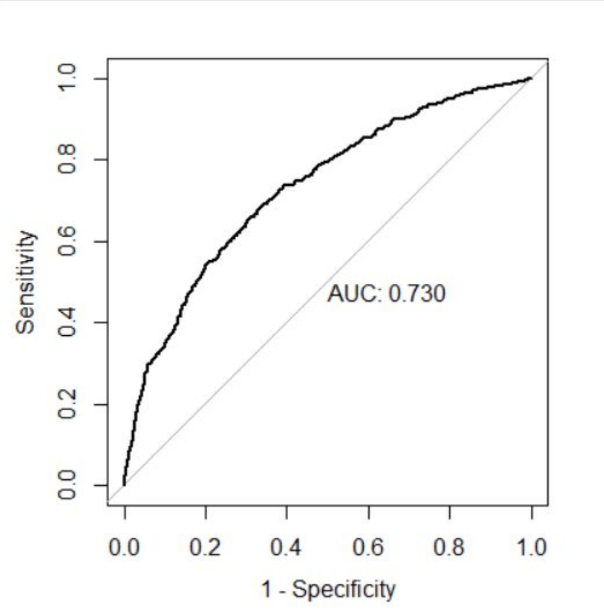


```
> xgb$finalModel
##### xgb.Booster
raw: 39.6 Kb
call:
  xgboost::xgb.train(params = list(eta = param$eta, max_depth = param$max_depth,
    gamma = param$gamma, colsample_bytree = param$colsample_bytree,
    min_child_weight = param$min_child_weight, subsample = param$subsample),
    data = x, nrounds = param$nrounds, objective = "binary:logistic")
params (as set within xgb.train):
  eta = "0.4", max_depth = "2", gamma = "0", colsample_bytree = "0.8", min_child_weight = "1", subsample = "1", objective = "binary:
logistic", silent = "1"
xgb.attributes:
  niter
callbacks:
  cb.print.evaluation(period = print_every_n)
# of features: 10
niter: 100
nfeatures : 10
xNames : Business_CodeCANADA Business_CodeEMEA Business_CodeUSA total_products total_transactions total_accounts total_revenue total
_usage Employee_Count_Total Number_of_Family_Members
problemType : Classification
tuneValue :
           nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
89     100         2 0.4     0              0.8                1         1
obsLevels : Yes No
param :
        list()
```





```
Confusion Matrix and Statistics

          Reference
Prediction  Yes   No
       Yes  229  916
       No   251 4645

               Accuracy : 0.8068
                 95% CI : (0.7966, 0.8167)
    No Information Rate : 0.9205
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1913

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

            Sensitivity : 0.47708
            Specificity : 0.83528
         Pos Pred Value : 0.20000
         Neg Pred Value : 0.94873
             Prevalence : 0.07946
         Detection Rate : 0.03791
   Detection Prevalence : 0.18954
      Balanced Accuracy : 0.65618

       'Positive' Class : Yes
```


AUC: 0.730
Sensitivity
1 - Specificity