Case Study 4: Customer Acquisition & Retention

Lee Waters

4/18/2021

I. Executive Summary

The `acquisitionRetention` data set from the `SMCRM` library (Verbeke, 2016) is used to predict which customers are likely to be acquired. Three models are applied, including: logistic regression (LR), decision tree (DT), and random forest (RF). The full data set is partitioned for training and testing purposes. Each model is fit to the training data using the features relevant to `acquisition`: `acq_exp`, `acq_exp_sq`, `industry`, `revenue`, and `employees`. Logistic regression achieves the best performance with 0.82 accuracy.

A random forest is then fit to the full data set to isolate the customers that the model correctly predicts to be acquired (289 customers). This modified data frame is again split into train and test sets. The train set is used to evaluate possible interaction effects using variable importance. This shows significant interaction between `ret_exp_sq:profit` but its inclusion in the random forest model does not improve performance. Grid search is used to optimize the hyperparameters by comparing the error rate of the out-of-bag samples. The model is re-trained using these optimal parameters (mtry=3, nodesize=4, ntree=4000) and then used to make predictions on the test data. The mean squared error (MSE) is calculated as 1285 days and shows good performance from the predicted versus actual `duration` plot.

One drawback of random forest is its lack of interpretability. Partial dependence plots (PDP) are generated to overcome this issue. These plots show how each predictor influences the response variable when the other variables are held constant.

II. The Problem

Customer acquisition and retention is a key focus of any business. The first step is to predict which customers are likely to be acquired. With this information, a business can choose to ignore customers not likely to engage with the company. This saves the company time and money by not wasting their effort on uninterested parties. It also permits the company to alter their strategy to try to acquire customers not likely to join with the current strategy.

The second step is to better understand the customers of your business. By predicting customer `duration`, a company can predict which customers have a high probability of ending their relationship with the firm. The company can then target these high-risk customers with incentives, such as pricing offers or direct messages, to increase their likelihood of staying.

The purpose of this study is to develop models to accurately predict customer `acquisition` and `duration`. This is important because it decreases the cost of the marketing campaign and uses company resources more efficiently. This study will answer two critical questions: 1) which customers are likely to be acquired and 2) how long are these customers likely to stay?

This report will first discuss the related literature as it pertains to building models to predict customer acquisition and retention. It will then discuss the methodology used in this study. A discussion of the data will follow, including the necessary pre-processing steps to prepare the data for analysis. The results will then be presented, including a comparison of the three customer acquisition models (logistic regression, decision tree, and random forest) and the customer duration model (random forest). The latter will only include customers that are accurately predicted by the classification model. It will also be optimized by exploring possible interaction effects and hyperparameters. The final model will be evaluated by making predictions on test data and calculating mean squared error. Finally, partial dependence plots will be generated to aid interpretability of the final model.

III. Review of Related Literature

The literature pertaining to customer acquisition and retention models is well established. In one study, ten analytical techniques that belong to different categories of learning are compared based on their performance for churn prediction problems (Sabbeh, 2018). These techniques include: Discriminant Analysis, Decision Trees (CART), instance-based learning (k-nearest neighbors), Support Vector Machines, Logistic Regression, ensemble–based learning techniques (Random Forest, Ada Boosting trees and Stochastic Gradient Boosting), Naïve Bayesian, and Multi-layer perceptron. The models were fit to telecommunications data consisting of 3333 records and random forest and ADA boost achieved the best performance with 96% accuracy.

Sabbeh investigates customer churn without acquisition. Unfortunately, these are not independent processes and can lead to biased and misleading results. Thomas (2001) presents a model that estimates the length of a customer's lifetime and adjust for this bias. The author's results show the financial impact of not accounting for the effect of acquisition on customer retention.

While this study uses machine learning techniques covered in other research (Sabbeh, 2008), it has a 6x smaller sample size which impacts results. This work demonstrates that other modelling approaches can achieve superior performance when the number of observations is small. It also bridges the two modelling frameworks (customer acquisition and retention) building on the work presented by Thomas (2001).

IV. Methodology

The objective of this study is to use `acquisitionRetention` to predict which customers will be acquired and for how long. Customer `acquisition` will be predicted using logistic regression, decision tree, and random forest classification. The accuracy of each model will be assessed by tabulating a confusion matrix. Customer `duration` will then be predicted using random forest for the customers accurately identified by a random forest model. Variable importance will be computed to detect interactions and optimize hyperparameters. Partial dependence plots will also be generated to increase interpretability of the final model.

Each approach has its own methodological assumptions and limitations. The simplest method, logistic regression, assumes a binary response, a linear relationship between the logit of the outcome and each predictor variable, no influential points, and no multicollinearity. Decision tree is a non-parametric method that has no assumptions about the space distribution and the classifier structure. However, DT models are prone to over-fitting and are not robust. Random forest aggregates many decision trees and improves predictive performance at the cost of interpretability.

The `acquisitionRetention` data is loaded into R from the `SMCRM` library. The data frame consists of 500 observations and 15 variables.

These include:

- `customer`: customer number (from 1 to 500)
- `acquisition`: 1 if the prospect was acquired, 0 otherwise
- `duration`: number of days the customer was a customer of the firm, 0 if `acquisition=0`
- `profit`: customer lifetime value (CLV) of a given customer (`-acq_exp` if `acquisition=0`)
- `acq_exp`: total dollars spent on trying to acquire this prospect
- `ret_exp`: total dollars spent on trying to retain this customer
- `acq_exp_sq`: square of the total dollars spent on trying to acquire this prospect
- `ret_exp_sq`: square total dollars spent on trying to retain this customer
- `freq`: number of purchases the customer made during that customer's lifetime with the firm (0 if `acquisition=0`)
- `freq_sq`: square of the number of purchases the customer made during that customer's lifetime with the firm
- `crossbuy`: number of product categories the customer purchased from during that customer's lifetime with the firm (0 if `acquisition=0`)
- `sow`: share-of-wallet, percentage of purchases the customer makes from the given firm given the total amount of purchases across all firms in that category
- `industry`: 1 if the customer is in the B2B industry, 0 otherwise
- `revenue`: annual sales revenue of the prospect's firm (in millions of dollar)

- `employees`: number of employees in the prospect's firm

All the features are initially numerical, but `acquisition` and `industry` are converted to factors. Since there are only 500 observations, the full data set is used, and no sampling techniques are implemented.

V. Data

Data preprocessing is performed to prepare the data set for analysis. The data is first checked for missing values, but none are found. Therefore, no imputation or removal of observations is required.

```
# Check for missing values
sapply(df, function(x) sum(is.na(x)))
```

```
##    customer acquisition    duration      profit     acq_exp     ret_exp
##           0           0           0           0           0           0
##  acq_exp_sq  ret_exp_sq        freq     freq_sq    crossbuy         sow
##           0           0           0           0           0           0
##    industry     revenue   employees
##           0           0           0
```

A heatmap is produced (*ggplot2,* 2020) to check for multicollinearity *(Figure 1)*. This is relevant to the logistic regression portion of the analysis. Only a subset of the features is used because some features are not known prior to customer `acquisition`. For example, `freq` is not applicable when predicting whether a customer will be acquired. If features like this are used, the model will achieve perfect separation and unrealistic classification performance.

For the classification portion of this analysis, the following predictors are used:

- `acq_exp`
- `acq_exp_sq`
- `industry`
- `revenue`
- `employees`

These predictors are checked for multicollinearity, except for `industry`, which is categorical. The results show multicollinearity between `acq_exp` and `acq_exp_sq`. This can affect the logistic regression, because it violates one of its underlying assumptions. However, both predictors are retained in the model because the focus of the `acquisition` analysis is to *compare* the baseline LR, DT, and RF models, not *optimize* them.

The data is also checked for possible imbalance *(Figure 2)*. The plot shows that the number of customers acquired are approximately double their counterpart. This imbalance will also impact the results of the logistic regression because the model will be biased towards acquired customers. One option is to resample the data to achieve a more even split. Another option is to use the sample proportion for the LR cutoff which is the method implemented in this study. The same predictors checked for multicollinearity are also inspected for influential points, but none are found *(Figure 3).*

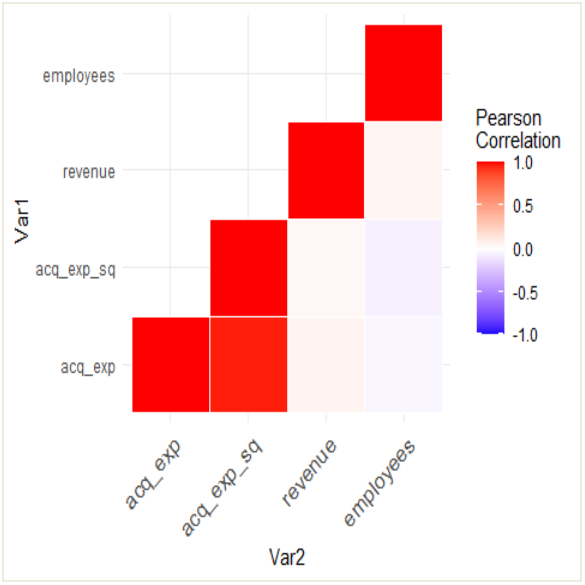


Figure 1 Heatmap of Pearson correlation

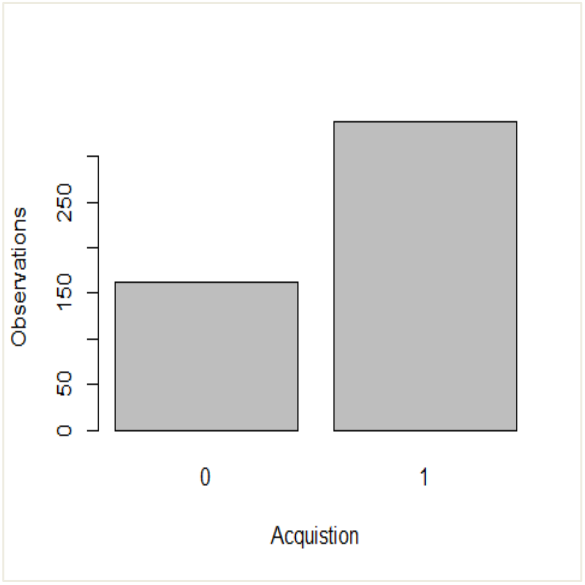


Figure 2 Plot of acquisition

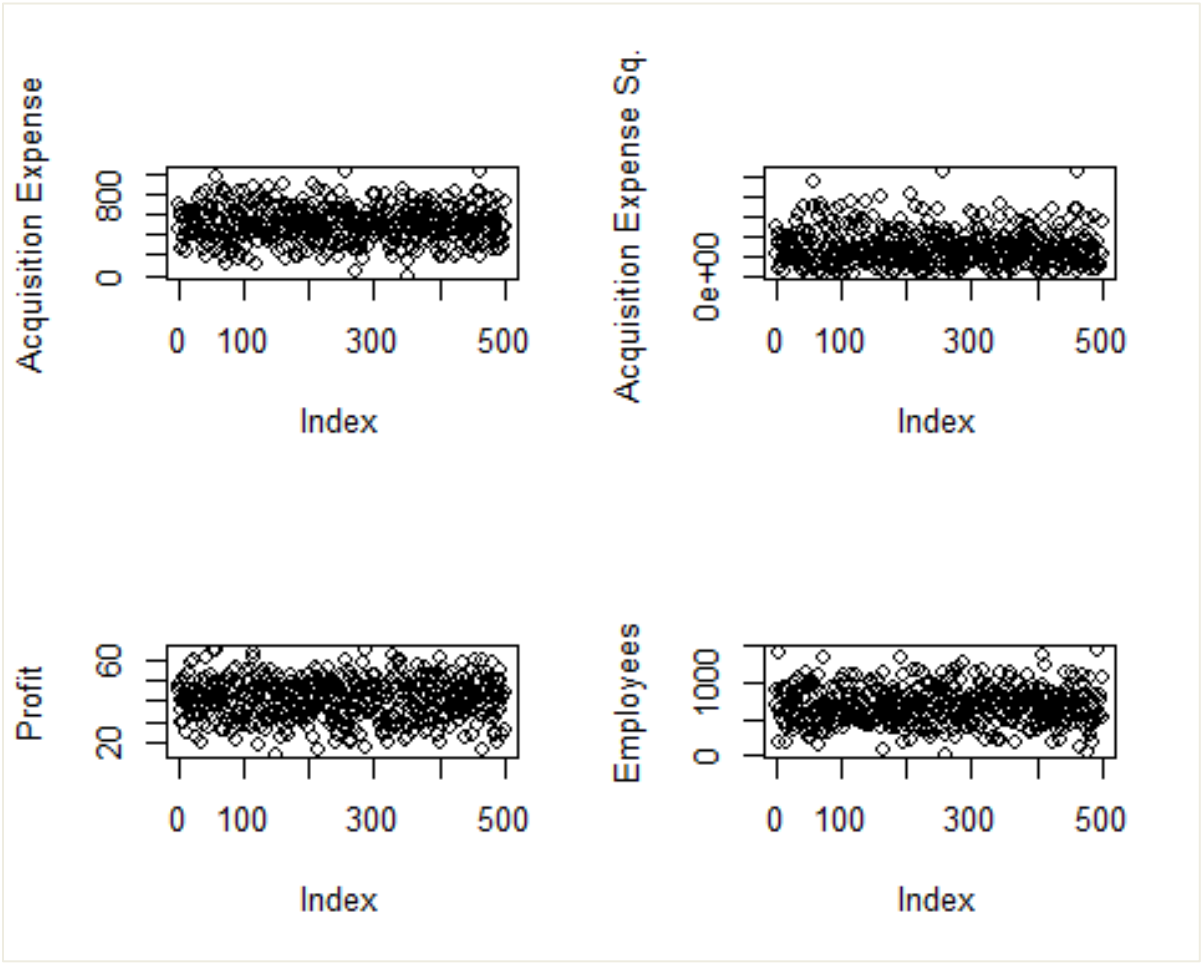


Figure 3 Plots of features

Finally, the data is partitioned into train and test sets to properly evaluate the model. The train data set has 400 observations, whereas the test set has 100. The train set will be used for fitting the classification models and the test set will be used to evaluate the models.

```
set.seed(42)
index = sample(nrow(df), .8*nrow(df))
train = df[index, ]
test = df[-index, ]

dim(train)

## [1] 400  15

dim(test)

## [1] 100  15
```

VI. Findings (Results)

Classification – Logistic Regression

A logistic regression is used to predict customer `acquisition`. The model is fit to the training data using `acq_exp`, `acq_exp_seq`, `industry`, `revenue`, and `employees` as features. Predictions are made on the test data with the sample proportion (0.68) used as a cutoff. A confusion matrix is generated, and the accuracy of the model is 0.82.

```
## Call:  glm(formula = acquisition ~ acq_exp + acq_exp_sq + industry +
##     revenue + employees, family = binomial, data = train)
##
## Coefficients:
## (Intercept)      acq_exp   acq_exp_sq    industry1      revenue    employees
##  -1.341e+01    2.776e-02   -2.831e-05    2.093e+00    6.836e-02    7.332e-03
##
## Degrees of Freedom: 399 Total (i.e. Null);  394 Residual
## Null Deviance:       501.5
## Residual Deviance: 285.3     AIC: 297.3
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 28 12
##          1  6 54
##
##                Accuracy : 0.82
##                  95% CI : (0.7305, 0.8897)
##     No Information Rate : 0.66
##     P-Value [Acc > NIR] : 0.0003021
##
##                   Kappa : 0.6154
##
##  Mcnemar's Test P-Value : 0.2385928
##
##             Sensitivity : 0.8182
##             Specificity : 0.8235
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.7000
##              Prevalence : 0.6600
##          Detection Rate : 0.5400
##    Detection Prevalence : 0.6000
##       Balanced Accuracy : 0.8209
##
##        'Positive' Class : 1
##
```

Classification – Decision Tree

A decision tree is fit to the train data using the same features as the LR. The variables used in tree construction are `employees`, `revenue`, `industry`, and `acq_exp` and the tree has 18 terminal nodes. Cross-validation is performed to see if the tree requires any pruning and the minimum deviance occurs at size 3. The decision tree is pruned to this size and re-fit to the data. The trained model is used to predict `acquistion` using the test data. A confusion matrix is generated, and the accuracy of the model is 0.76.

```
## Classification tree:
## tree(formula = acquisition ~ acq_exp + acq_exp_sq + industry +
##     revenue + employees, data = train)
## Variables actually used in tree construction:
## [1] "employees" "revenue"   "industry"  "acq_exp"
## Number of terminal nodes:  18
## Residual mean deviance:  0.6542 = 249.9 / 382
## Misclassification error rate: 0.1475 = 59 / 400
```

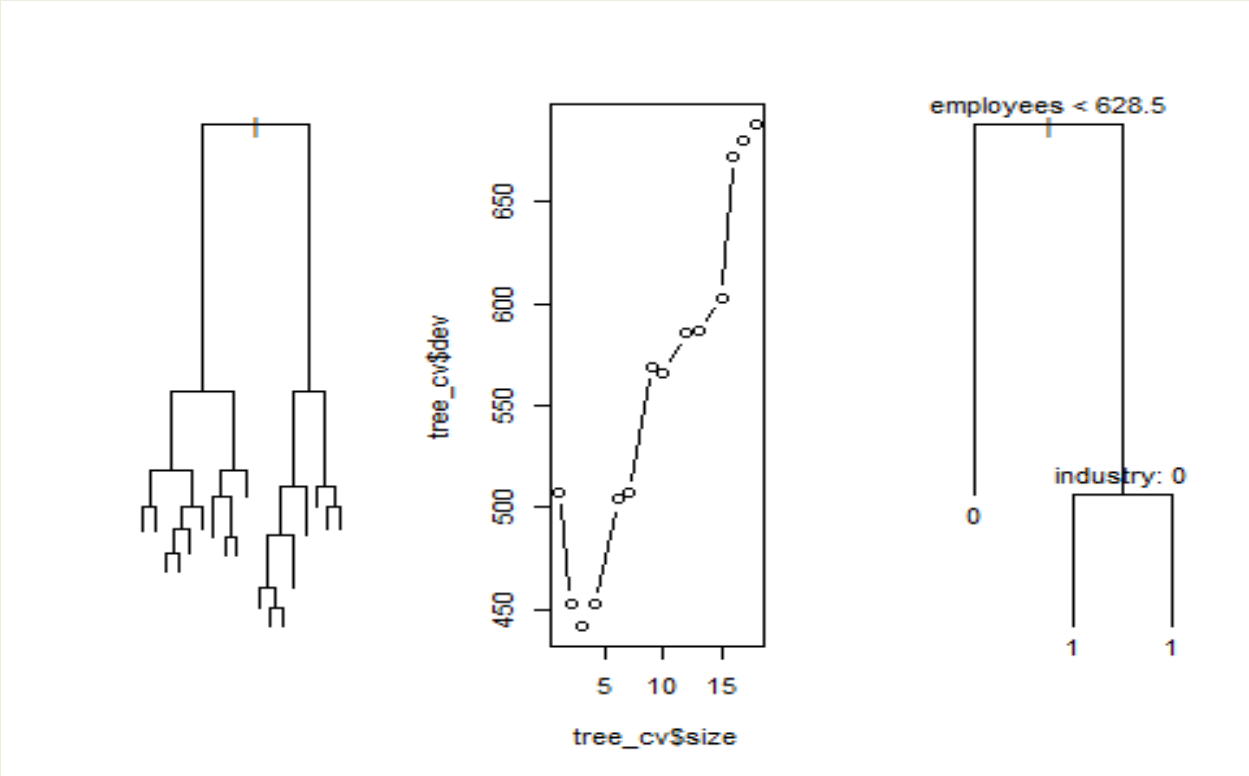


Figure 4 Decision tree, cross-validation plot, pruned tree

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
```

```
##          0 31 21
##          1  3 45
##
##                Accuracy : 0.76
##                  95% CI : (0.6643, 0.8398)
##     No Information Rate : 0.66
##     P-Value [Acc > NIR] : 0.0202816
##
##                   Kappa : 0.5261
##
##  Mcnemar's Test P-Value : 0.0005202
##
##             Sensitivity : 0.6818
##             Specificity : 0.9118
##          Pos Pred Value : 0.9375
##          Neg Pred Value : 0.5962
##              Prevalence : 0.6600
##          Detection Rate : 0.4500
##    Detection Prevalence : 0.4800
##       Balanced Accuracy : 0.7968
##
##        'Positive' Class : 1
##
```

Classification – Random Forest

A random forest is fit to the training data using the same features as LR and DT. The default number of trees (ntree=500) and the number of variables randomly selected as candidates at each split (mtry=2) is used. The trained model is used to make predictions on the test data. A confusion matrix is generated, and the accuracy of the model is 0.79.

```
## Call:
##  randomForest(formula = acquisition ~ acq_exp + acq_exp_sq + industry +
revenue + employees, data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 22.5%
## Confusion matrix:
##    0   1 class.error
## 0 76  52   0.4062500
## 1 38 234   0.1397059
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 22  9
##          1 12 57
##
##                Accuracy : 0.79
##                  95% CI : (0.6971, 0.8651)
##     No Information Rate : 0.66
##     P-Value [Acc > NIR] : 0.00318
##
##                   Kappa : 0.5219
##
##  Mcnemar's Test P-Value : 0.66252
##
##             Sensitivity : 0.8636
##             Specificity : 0.6471
##          Pos Pred Value : 0.8261
##          Neg Pred Value : 0.7097
##              Prevalence : 0.6600
##          Detection Rate : 0.5700
##    Detection Prevalence : 0.6900
##       Balanced Accuracy : 0.7553
##
##        'Positive' Class : 1
##
```

Logistic regression achieves the highest accuracy. Additional analysis would involve grid search and selecting the best hyperparameters. For example, the accuracy of random forest is improved by increasing mtry to the full feature set (i.e., bagging). Classification threshold can also be optimized based on sensitivity and specificity scores and the desired business outcome. However, these methods are outside the scope of this study and random forest is used for the regression portion of this analysis.

Regression – Random Forest

The random forest model is applied to the entire data set, because it had previously been applied to just the training data.

```
## Call:
##  randomForest(formula = acquisition ~ acq_exp + acq_exp_sq + industry +
revenue + employees, data = df)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 22.8%
## Confusion matrix:
##    0   1 class.error
## 0 97  65   0.4012346
## 1 49 289   0.1449704
```

A new data frame is constructed consisting of the customers that the RF accurately predicts to be acquired. This consists of 289 customers which is then split into new train and test sets. The train data set has 231 observations, whereas the test set has 58. The train set will be used for fitting the regression model and the test set will be used to evaluate the model.

```
df$rf2 = rf2$predicted
df2 = df[df$acquisition==1 & df$rf2==1,]
dim(df2)

## [1] 289  16
```

```
set.seed(42)
index = sample(nrow(df2), .8*nrow(df2))
train = df2[index, ]
test = df2[-index, ]


dim(train)

## [1] 231  16

dim(test)

## [1] 58 16
```

A random forest is fit to the training data. Different features are used for the regression analysis compared to the classification.

These include:

- profit
- acq_exp
- ret_exp
- acq_exp_sq
- ret_exp_sq
- freq
- freq_sq
- crossbuy
- sow
- industry
- revenue
- employees

Variable importance is used to identify any possible interaction terms. The error rate on the full tree is 1708 and there is significant interaction with ~~with~~ a difference of 4745 between ret_exp_sq:profit. The interaction between ret_exp_sq:ret_eq is ignored because it is the same feature squared. The ret_exp_sq:profit term is added, and another random forest model is trained to see if it improves performance. The error rate is the same and the first random forest model, without interaction effects, is used.

Original model

```
rfsrc(duration~profit+acq_exp+ret_exp+acq_exp_sq+ret_exp_sq+freq+freq_sq+cros
sbuy+sow+industry+revenue+employees, data=train, importance=T))
```

```
##                          Sample size: 231
##                     Number of trees: 1000
##           Forest terminal node size: 5
##       Average no. of terminal nodes: 27.963
## No. of variables tried at each split: 4
##              Total no. of variables: 12
##       Resampling used to grow trees: swor
##     Resample size used to grow trees: 146
##                             Analysis: RF-R
##                               Family: regr
##                     Splitting rule: mse *random*
##       Number of random split points: 10
##                 % variance explained: 96.22
##                           Error rate: 1707.92
```

Interaction

```
##                                Method: vimp
##                      No. of variables: 12
##            Variables sorted by VIMP?: TRUE
##    No. of variables used for pairing: 12
##     Total no. of paired interactions: 66
##              Monte Carlo replications: 1
##     Type of noising up used for VIMP: permute
##
##                             Var 1      Var 2     Paired   Additive Difference
## ret_exp_sq:ret_exp    14986.2617 13602.4738 38041.6341 28588.7355  9452.8987
## ret_exp_sq:profit     14986.2617  5105.5218 24836.3321 20091.7834  4744.5486
## ret_exp_sq:freq_sq    14986.2617   553.7556 15553.4749 15540.0172    13.4577
## ret_exp_sq:freq       14986.2617   562.6432 15490.2907 15548.9049   -58.6142
## ret_exp_sq:acq_exp    14986.2617   143.6796 15227.3395 15129.9413    97.3983
```

Interaction model

```
rfsrc(duration~profit+acq_exp+ret_exp+acq_exp_sq+ret_exp_sq+freq+freq_sq+cros
sbuy+sow+industry+revenue+employees+ret_exp_sq:profit, data=train))
```

```
##                            Sample size: 231
##                         Number of trees: 1000
##              Forest terminal node size: 5
##          Average no. of terminal nodes: 27.963
## No. of variables tried at each split: 4
##                  Total no. of variables: 12
##          Resampling used to grow trees: swor
##       Resample size used to grow trees: 146
##                               Analysis: RF-R
##                                 Family: regr
##                         Splitting rule: mse *random*
##          Number of random split points: 10
##                   % variance explained: 96.22
##                             Error rate: 1707.92
```

The hyperparameters are optimized by performing grid search and calculating the out-of-bag error rates (Routh, 2021).

The hyperparameter grid is constructed as follows:

- mtry (4, 5, 6)
- nodesize (4, 6, 8)
- ntree (4000, 5000, 6000)

This optimal model has mtry of 6, nodesize of 4, and ntree of 4000 and is used to make predictions on the test data.

```
# Identify optimal set of hyperparmeters based on OOB error
opt_i = which.min(oob_err)
print(hyper_grid[opt_i,])
```

```
##   mtry nodesize ntree
## 3    6        4  4000
```

The model is re-trained on the entire training data and predictions are made on the test data. The mean squared error is calculated as 1539 days. A plot of actual versus predicted duration shows strong linear correlation indicating a good result *(Figure 5).*

```
## Call:
##  randomForest(formula = duration ~ profit + acq_exp + ret_exp +
acq_exp_sq + ret_exp_sq + freq + freq_sq + crossbuy + sow +      industry +
revenue + employees, data = train, mtry = 6, nodesize = 4,      ntree = 4000)
##                Type of random forest: regression
##                      Number of trees: 4000
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 1498.631
##                     % Var explained: 96.67
```

```
# Compute val error
(mean((yhat_rf-test$duration)^2))
```

```
## [1] 1284.756
```

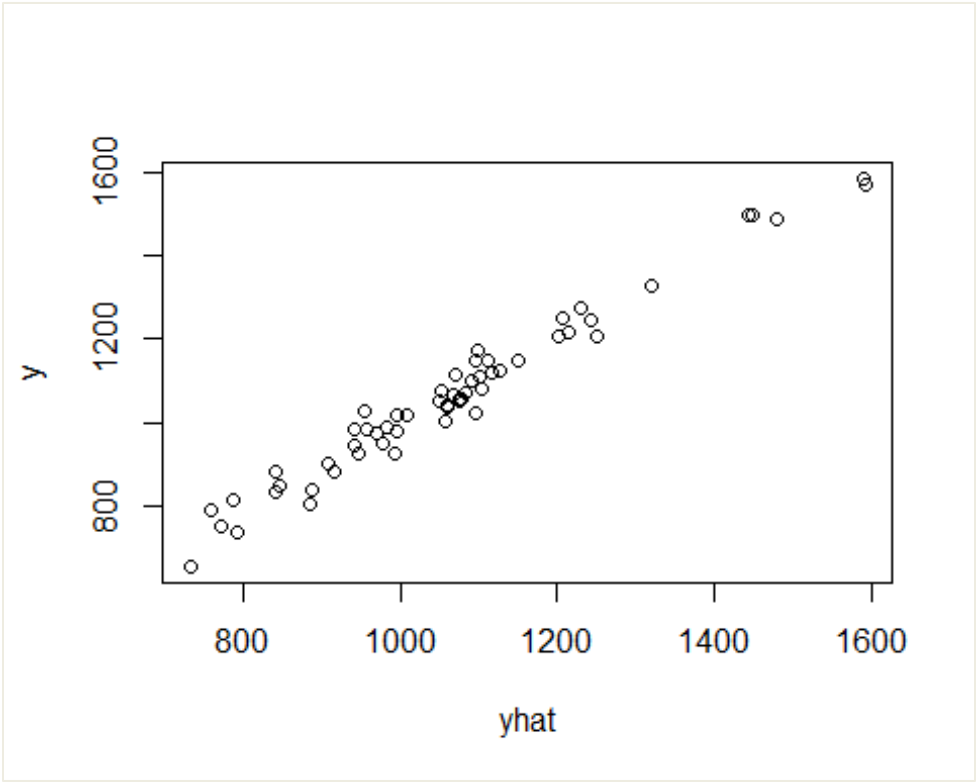
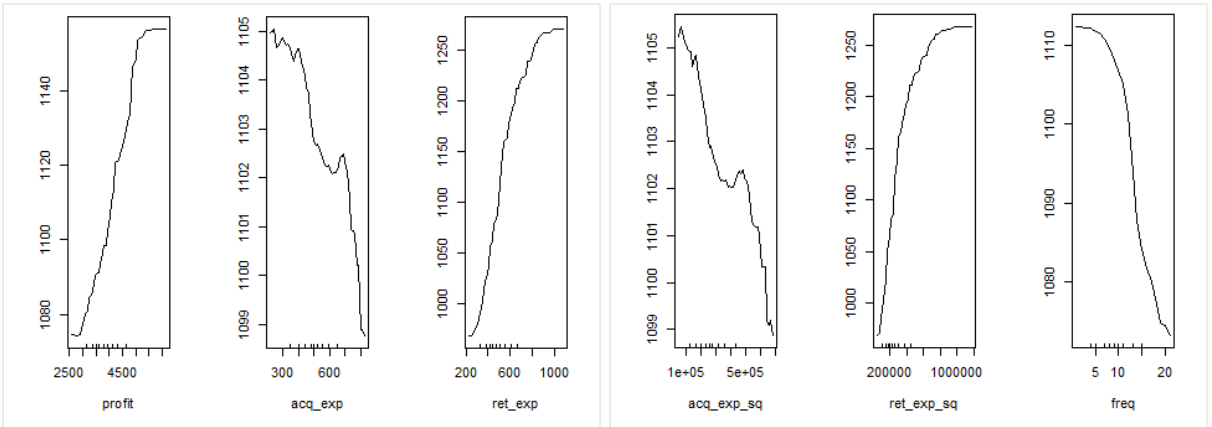


Figure 5 Actual versus predicted duration

Partial dependent plots are generated for each feature to interpret the final model. These plots are like the coefficients in a linear or logistic regression. For example, the PDP of profit shows that when the other variables are held constant, duration increases with profit. These plots are useful for improving the interpretability of a random forest models, which is one of their chief drawbacks.



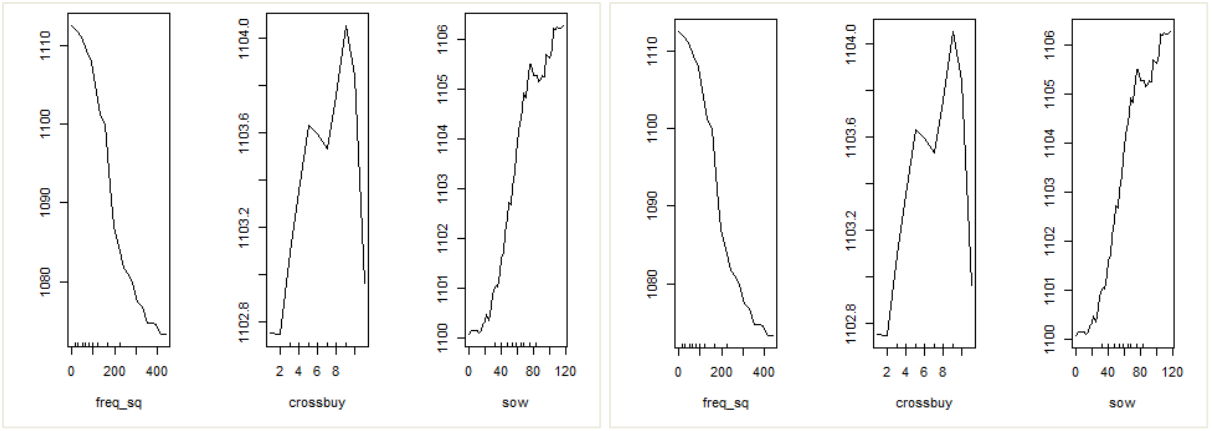


Figure 6 Partial dependence plots of final model

VII. Conclusions and Recommendations

Logistic regression, decision tree, and random forest are used to predict `acquisition` from the `acquisitionRetention` data set. Logistic regression achieves the best performance with 0.82 accuracy.

Random forest is then fit to the entire data set to subset customers that the model accurately predicts to be acquired. The model is optimized to predict `duration` and achieves a MSE of 1285 days.

There are several opportunities to improve the results of this study. The logistic regression had the highest classification accuracy, but the random forest model was used to subset the data. This indicates that there is opportunity to optimize the classification model, using similar techniques as the regression analysis. This would increase the number of customers that the model accurately predicts to be acquired, increase the number of training samples, and therefore improve the results of the `duration` model. Two additional opportunities to improve performance include exploring different interaction effects and expanding the grid search matrix.

Finally, boosting is another ensemble learning technique like random forest. Whereas random forest reduces variance to prevent over-fitting, boosted trees primarily reduce bias by converting weak learners to strong ones. This can improve the predictive performance of the customer `duration` model and is another avenue that future authors can explore.

VIII. Appendix

References

Verbeke, Tobias (2016). *SMCRM: Data Sets for Statistical Methods in Customer Relationship Management by Kumar and Petersen (2012).* Accessed April 12, 2021. https://CRAN.R-project.org/package=SMCRM.

Sabbeh, Sahar (2018). *Machine-Learning Techniques for Customer Retention: A Comparative Study. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 2, 2018.*

Thomas, Jacquelyn S. (2001). *A Methodology for Linking Customer Acquisition to Customer Retention. Journal of Marketing Research.*

ggplot2: Quick correlation matrix heatmap - R software and data visualization (2020). Accessed March 30, 2021. http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization

Routh, Pavall. *Random Forest.* Accessed April 13, 2021. https://utsa.blackboard.com/.