Casestudy 4 – Customer Acquisition

Vishwa Koppisetti

February 22, 2019

I. Executive Summary

The customer acquisition dataset contains customer data like their acquisition expense, retention expense, customer lifetime value, duration, frequency of purchase, number of categories of purchase, etc.
This case study is to analyze the following:

- Use customer Acquisition data set to predict which customers will be acquired based on a feature set using a random forest.
- Compute variable importance to detect interactions and optimize hyper-parameters
- Compare the accuracy of model with a decision trees and logistic regression model
- Compare the accuracy improvements if any for portfolios of CLV

They were several variables provided for customer acquisition analysis but most of them were information about the customer once he/she is acquired. So removed all such variables from the models and used variables revenue, censor, employees, industry, acq_expense for predicting customer acquisition.

All the variables provided were of type numeric so converted acquisition, censor and industry to factor as required.

Decision Tree: We run decision tree with acquisition as response on training data that has 350 observations. The most importance variable was acq_expense which was the first node that was split. The prediction was done on test set of 150 observations and the accuracy was 93%

This model helps in easy interpretation but has risk of overfitting the data that could be overcome by random forest

Random Forest:

We run Random Forest with same explanatory variables and acquisition as response on training data. The model is run without tuning the hyper parameters and When predicted on test set, we get an accuracy of 94% which is slightly better than decision tree model.

Next, we perform a grid search to find the optimal ntree and mtry values with which we run the random forest model. The accuracy is 94.6% which is better than the model without optimal values.

Logistic Regression:

We run logistic regression to model acquisition with acq_expense, revenue and employees on training data. The other variables were removed from the model because they were insignificant. When predicted on test set, we get an accuracy of 88.6%. Logistic regression has the lowest accuracy compared to all other models run so far. Logistic regression runs faster and interprets better but is less predictive

We then check the portfolio of CLV based on the prediction accuracies of customers in the test set. Average and total acquisition expense for top 25% customers are $553 and $21024 respectively. The next expense category is top 75% customer with average and total acquisition expense $561 and $20770 respectively.

We check the portfolio of CLV based on the prediction accuracies of customers in the test set. Average and total acquisition expense for top 25% customers are $553 and $21024 respectively. The next expense category is top 75% customer with average and total acquisition expense $561 and $20770 respectively. Tibble shown in Appendix G

With the help of ggplot, we compare the accuracy improvements for portfolios of CLV. Appendix G

Linear model has the highest error rate for all customers given that it is not recommended method for binary response. Then Logistic has the next highest error. For random forest without optimal hyper parameters error rate was more than model with optimal hyper parameters. The top 25% customers higher error rates compared to top 75% customers. So, the accuracy has improved as we predicted for larger group of customers.

The most important variables were acq_expense, clv and censor but the interaction between acq_expense, clv was not significant.

I recommend random forest that was modelled with optimal hyper parameters because it has highest accuracy of 94.6% and it reduces risk of overfitting, faced by decision trees, by including randomness in variable selection and bootstrap sampling.

II. The Problem

A. Introduction/Background

Customer acquisition refers to gaining new consumers. Acquiring new customers involves persuading consumers to purchase a company's products and/or services. Companies and organizations consider the cost of customer

acquisition as an important measure in evaluating how much value customers bring to their businesses.

Customer acquisition management refers to the set of methodologies and systems for managing customer prospects and inquiries that are generated by a variety of marketing techniques.

The customer acquisition dataset from the SMCRM package : Data Sets for Statistical Methods in Customer Relationship Management by Kumar and Petersen (2012).has 500 observations and 16 variables.

B. Purpose of study/importance of study/statement of problem

- Use customer Acquisition data set to predict which customers will be acquired based on a feature set using a random forest.
- Compute variable importance to detect interactions and optimize hyper-parameters
- Compare the accuracy of model with a decision trees and logistic regression model
- Compare the accuracy improvements if any for portfolios of CLV

C. Questions to be answered/conceptual statement of hypotheses

- Which of the three; a logit model, decision tree or random forest gives most accurate predictions
- Which customers will be acquired based on a feature set using a random forest

D. Outline of remainder of report (brief)

- Procedure followed to model the response variable on train set using for the different models.
- Assessing the performance of the model using test set
- Comparing accuracies of models
- Comparing accuracy improvements if any for portfolio of CLV

III. Review of Related Literature

A. Acquaint reader with existing methodologies used in this area.

Methodologies such as logistic regression can be used when speed and interpretability is preferred over predictive power. Random forest is used over decision trees to avoid over fitting.

IV. Methodology

A. Identification, classification and operationalization of variables.

- The customer acquisition dataset from the SMCRM package : Data Sets for Statistical Methods in Customer Relationship Management by Kumar and Petersen (2012).has 500 observations and 16 variables.
- 350 observations of train set, 150 observations of test set.
- Most of the variables have information about customer once the customer is acquired so we have 7 variables of interest with a binary dependent variable and 6 numeric/binary independent variables. Classification and operationalization in Appendix A

B. Statements of hypotheses being tested and/or models being developed.

- To find a highly accurate model among Decision Tree model, a logit model, and random forest for prediction customer acquisition
- Which customers will be acquired based on a feature set using a random forest

C. Sampling techniques, if full data is not being used.

The customer acquisition dataset was split into 70% train set of 350 observations and 30% test set of 150 to make predictions. Appendix B

D. Data collection process, including data sources, data size, etc. Primary/secondary?

- For this case analysis, we will use a customer acquisition dataset from SMCRM package : Data Sets for Statistical Methods in Customer Relationship Management by Kumar and Petersen (2012). secondary data. It consists of data 500 observations and 16 variables

The dependent variable for the analysis is acquisition which is binary – acquired or not. They are several independent variables which

E. Modeling analysis/techniques used

Different modelling techniques such as a logit model, Decision tree and Random Forests are used to determine which model provides good accuracy

Appendix C-F

F. Methodological assumptions and limitations.

Decision trees are non-parametric methods that do not require assumptions to be met. They are easy to understand and useful in data exploration. It can handle numerical and categorical data but has it own limitations of overfitting of data. It is not robust method. Uncertainties in data could have significant impact on the decisions.

Logistic regression runs faster and interprets better but is less predictive and does not work well for unbalanced data when compared to random forest

Random forest can take time with hypergrid for optimal ntree and mtry but it gives better prediction accuracy.

V. Data

A. Data cleaning Appendix A

There are no missing values in any of the datasets.

The variables having information after the customer is acquired are filtered out as they are not helpful to make predictions on customer acquisition

B. Data preprocessing Appendix A-B

Converted acquisition, industry and censor to factors as required.

The customer acquisition dataset was split into 70% train set of 350 observations and 30% test set of 150 to make predictions

C. Data Limitations

There were no missing values in the dataset, but the number of observations is less, and data is unbalanced in terms of response class. Most of the variables contained information after the customer is acquired are filtered out as they are not helpful to make predictions on customer acquisition. This led to fewer predictors than already available.

VI. Findings (Results)

A. Results presented in tables or charts when appropriate

The variable importance plot shown in Appendix D shows that acq_expense , clv and censor are the most important variables.

We check the portfolio of CLV based on the prediction accuracies of customers in the test set. Average and total acquisition expense for top 25% customers are $553 and $21024 respectively. The next expense category is top 75% customer with average and total acquisition expense $561 and $20770 respectively. Tibble shown in Appendix G

With the help of ggplot, we compare the accuracy improvements for portfolios of CLV. Appendix G

Linear model has the highest error rate for all customers given that it is not recommended method for binary response. Then Logistic has the next highest error. For random forest without optimal hyper parameters error rate was more than model with optimal hyper parameters. The top 25% customers higher error

rates compared to top 75% customers. So, the accuracy has improved as we predicted for larger group of customers.

B. Results reported with respect to hypotheses/models.

Decision Tree: We run decision tree with acquisition as response on training data that has 350 observations. The most importance variable was acq_expense which was the first node that was split. The prediction was done on test set of 150 observations and the accuracy was 93%

This model helps in easy interpretation but has risk of overfitting the data that could be overcome by random forest. Appendix C

Random Forest:

We run Random Forest with same explanatory variables and acquisition as response on training data. The model is run without tuning the hyper parameters and When predicted on test set, we get an accuracy of 94% which is slightly better than decision tree model.

Next, we perform a grid search to find the optimal ntree and mtry values with which we run the random forest model. The accuracy is 94.6% which is better than the model without optimal values. Appendix D-E

Logistic Regression: Appendix F

We run logistic regression to model acquisition with acq_expense, revenue and employees on training data. The other variables were removed from the model because they were insignificant. When predicted on test set, we get an accuracy of 88.6%. Logistic regression has the lowest accuracy compared to all other models run so far. Logistic regression runs faster and interprets better but is less predictive

VII. Conclusions and Recommendations

We check the portfolio of CLV based on the prediction accuracies of customers in the test set. Average and total acquisition expense for top 25% customers are $553 and $21024 respectively. The next expense category is top 75% customer with average and total acquisition expense $561 and $20770 respectively. Tibble shown in Appendix G

The customer with more acq_expense, customers to firms with more number of employees and more revenue are likely to be acquired

With the help of ggplot, we compare the accuracy improvements for portfolios of CLV. Appendix G

Linear model has the highest error rate for all customers given that it is not recommended method for binary response. Then Logistic has the next highest error. For random forest without optimal hyper parameters error rate was more than model with optimal hyper parameters. The top 25% customers higher error rates

compared to top 75% customers. So, the accuracy has improved as we predicted for larger group of customers.

The most important variables were acq_expense, clv and censor but the interaction between acq_expense, clv was not significant.

I recommend random forest that was modelled with optimal hyper parameters because it has highest accuracy of 94.6% and it reduces risk of overfitting, faced by decision trees, by including randomness in variable selection and bootstrap sampling.

Appendix A

```
str(customerAcquisition)
```

```
## 'data.frame':    500 obs. of  16 variables:
##  $ customer      : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ acquisition   : num  1 0 0 1 1 0 0 0 1 1 ...
##  $ first_purchase: num  434 0 0 226 363 ...
##  $ clv           : num  0 0 0 5.73 0 ...
##  $ duration      : num  384 0 0 730 579 0 0 0 730 730 ...
##  $ censor        : num  0 0 0 1 0 0 0 0 1 1 ...
##  $ acq_expense   : num  760 148 253 610 672 ...
##  $ acq_expense_sq: num  578147 21815 63787 371771 452068 ...
##  $ industry      : num  1 1 1 1 1 0 0 0 1 1 ...
##  $ revenue       : num  30.2 39.8 54.9 45.8 69 ...
##  $ employees     : num  1240 166 1016 122 313 ...
##  $ ret_expense   : num  2310 0 0 2193 801 ...
##  $ ret_expense_sq: num  5335130 0 0 4807451 641825 ...
##  $ crossbuy      : num  5 0 0 2 4 0 0 0 1 1 ...
##  $ frequency     : num  2 0 0 12 7 0 0 0 11 14 ...
##  $ frequency_sq  : num  4 0 0 144 49 0 0 0 121 196 ...
```

```
CA = customerAcquisition
summary(CA)
```

```
##     customer      acquisition    first_purchase       clv        
##  Min.   :  1.0   Min.   :0.000   Min.   :  0.0   Min.   :0.000  
##  1st Qu.:125.8   1st Qu.:0.000   1st Qu.:  0.0   1st Qu.:0.000  
##  Median :250.5   Median :1.000   Median :207.9   Median :0.000  
##  Mean   :250.5   Mean   :0.584   Mean   :217.5   Mean   :1.776  
##  3rd Qu.:375.2   3rd Qu.:1.000   3rd Qu.:397.8   3rd Qu.:5.270  
##  Max.   :500.0   Max.   :1.000   Max.   :731.6   Max.   :9.225  
##     duration         censor      acq_expense     acq_expense_sq   
##  Min.   :  0.0   Min.   :0.00   Min.   : 14.21   Min.   :   201.9  
##  1st Qu.:  0.0   1st Qu.:0.00   1st Qu.:359.95   1st Qu.:129567.7  
##  Median :188.0   Median :0.00   Median :515.31   Median :265544.4  
##  Mean   :299.9   Mean   :0.27   Mean   :513.90   Mean   :312876.4
```

```
##  3rd Qu.:730.0   3rd Qu.:1.00   3rd Qu.:685.38   3rd Qu.:469746.4
##  Max.   :730.0   Max.   :1.00   Max.   :968.06   Max.   :937140.2
##     industry       revenue        employees       ret_expense
##  Min.   :0.00   Min.   : 1.86   Min.   :   4.0   Min.   :   0.0
##  1st Qu.:0.00   1st Qu.:28.01   1st Qu.: 270.0   1st Qu.:   0.0
##  Median :1.00   Median :40.09   Median : 588.5   Median : 871.3
##  Mean   :0.59   Mean   :39.81   Mean   : 668.5   Mean   : 885.3
##  3rd Qu.:1.00   3rd Qu.:51.96   3rd Qu.:1025.5   3rd Qu.:1583.8
##  Max.   :1.00   Max.   :76.77   Max.   :1968.0   Max.   :2927.7
##  ret_expense_sq       crossbuy       frequency       frequency_sq
##  Min.   :      0   Min.   :0.000   Min.   : 0.000   Min.   :  0.0
##  1st Qu.:      0   1st Qu.:0.000   1st Qu.: 0.000   1st Qu.:  0.0
##  Median : 759283   Median :1.000   Median : 3.000   Median :  9.0
##  Mean   :1540477   Mean   :2.032   Mean   : 5.162   Mean   : 58.5
##  3rd Qu.:2508351   3rd Qu.:4.000   3rd Qu.:10.000   3rd Qu.:100.0
##  Max.   :8571662   Max.   :6.000   Max.   :16.000   Max.   :256.0

CA$acquisition = as.factor(CA$acquisition)
CA$censor = as.factor(CA$censor)
CA$industry = as.factor(CA$industry)
```

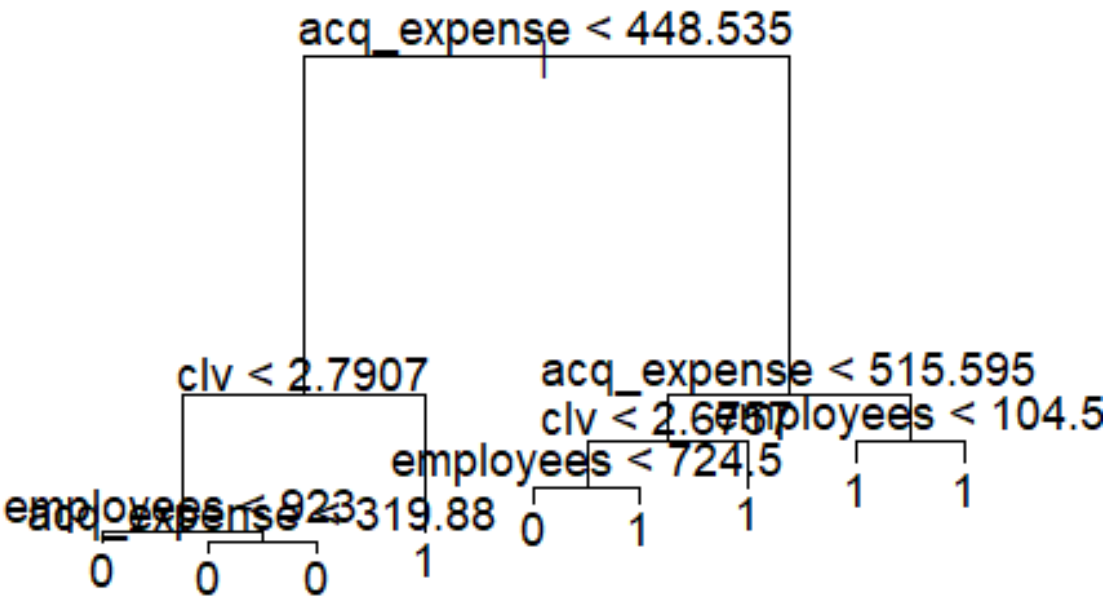
Appendix B

```
set.seed(123)
train = sample(1:nrow(CA),nrow(CA)*0.7,rep=FALSE)
test = -train
```

Appendix C

Decision Tree

```
dt.model <- tree(acquisition ~ clv+censor+acq_expense+industry+revenue+employ
ees,
                            data = CA[train,])


plot(dt.model)

text(dt.model, pretty = 0, cex= 1.1)
```



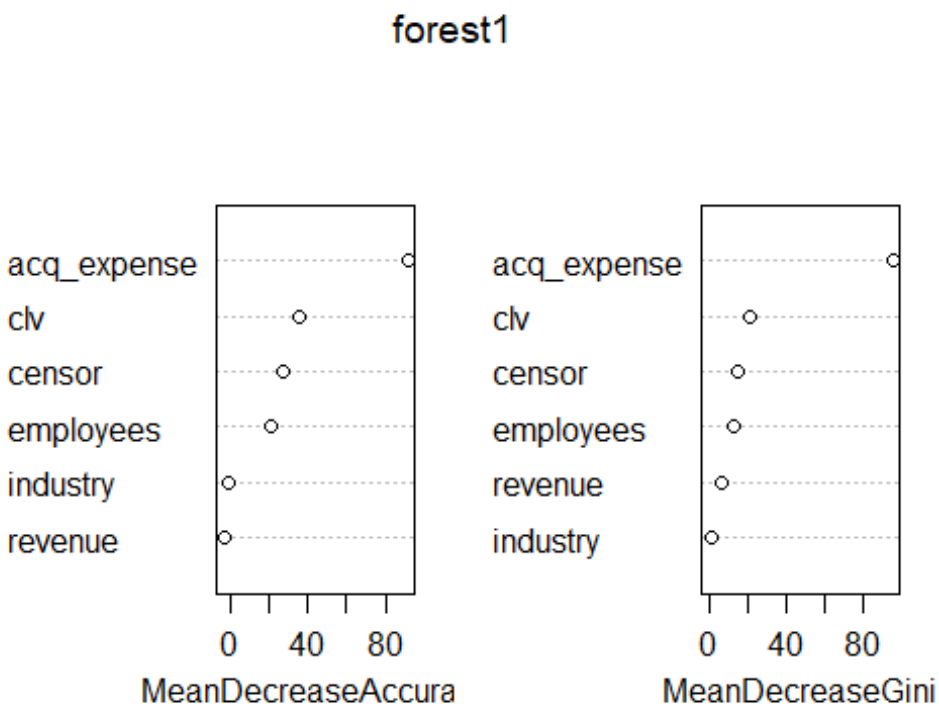

```
tree.pred = predict(dt.model, CA[test,], type = "class")
table(tree.pred,CA[test,]$acquisition)

##
## tree.pred  0  1
##         0 52  1
##         1  9 88

tree.acc = mean(tree.pred==CA[test,]$acquisition)*100;tree.acc

## [1] 93.33333
```

Appendix D

Random Forest without optimal parametres

```
set.seed(123)
forest1 <- randomForest(acquisition ~ clv+censor+acq_expense+industry+revenue
+employees, data = CA[train,],importance = T,
                        ntree = 1000)
forest1

##
## Call:
##  randomForest(formula = acquisition ~ clv + censor + acq_expense +      in
dustry + revenue + employees, data = CA[train, ], importance = T,      ntree
= 1000)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 3.71%
## Confusion matrix:
##     0   1 class.error
## 0 139   8  0.05442177
## 1   5 198  0.02463054

  forest1$err.rate[length(forest1$err.rate)]

## [1] 0.02463054

  forest1$err.rate[1000]

## [1] 0.03714286

  predRFun = predict(forest1, newdata = CA[test,], type="class")
  table(predRFun, CA[test,]$acquisition)

##
## predRFun  0  1
##        0 54  2
##        1  7 87

  accRFun = mean(predRFun == CA[test,]$acquisition)*100;accRFun

## [1] 94

forest1$importance

##                          0            1 MeanDecreaseAccuracy
## clv          0.1280516353  0.058723118         0.0874750397
## censor       0.0921956842  0.042452705         0.0632448923
## acq_expense  0.3502528629  0.201936413         0.2632751053
## industry     0.0006720551 -0.001173518        -0.0003490364
## revenue     -0.0010898670 -0.001786242        -0.0015166235
## employees    0.0335345442  0.005321187         0.0172022569
##             MeanDecreaseGini
```

```
## clv               21.0616458
## censor            15.2397353
## acq_expense       95.3286717
## industry           0.8761935
## revenue            6.5898011
## employees         13.0770886

varImpPlot(forest1)
```



Appendix E

Random Forest with optimal parameters

```
  # Grid search for opt mtry and ntree
mtry.values <- seq(4,6,1)

ntree.values <- seq(1e3,6e3,1e3)
  # Create an empty vector to store OOB error values
hyper_grid <- expand.grid(mtry = mtry.values,  ntree = ntree.values)

  oob_err <- c()
library(randomForest)
  # Write a loop over the rows of hyper_grid to train the grid of models
  for (i in 1:nrow(hyper_grid)) {
```

```
    # Train a Random Forest model
    set.seed(123)

    model <- rfsrc(acquisition ~ clv+censor+acq_expense+industry+revenue+empl
oyees, data = CA[train,], importance = T, mtry =                 hyper_grid$m
try[i],ntree = hyper_grid$ntree[i])

    # Store OOB error for the model
     oob_err[i]<-model$err.rate[length(model$err.rate)]

  }
  # Find location of opt index in grid search
  opt_i <- which.min(oob_err)
  print(hyper_grid[opt_i,])

##   mtry ntree
## 1    4  1000

  # Run random forest with opt mtry and ntree, predict on test set, and evalu
ate MSE.
  set.seed(123)
  resultsRF = randomForest(acquisition ~ clv+censor+acq_expense_sq+industry+r
evenue+employees, data = CA[train,], mtry = hyper_grid$mtry[opt_i],
                           ntree = hyper_grid$ntree[opt_i],importance = T)
  resultsRF

##
## Call:
##  randomForest(formula = acquisition ~ clv + censor + acq_expense_sq +
industry + revenue + employees, data = CA[train, ], mtry = hyper_grid$mtry[op
t_i],      ntree = hyper_grid$ntree[opt_i], importance = T)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 3.71%
## Confusion matrix:
##     0   1 class.error
## 0 139   8  0.05442177
## 1   5 198  0.02463054

  predRF = predict(resultsRF, newdata = CA[test,], type="class")
  table(predRF, CA[test,]$acquisition)

##
## predRF  0  1
##      0 54  1
##      1  7 88

  accRF = mean(predRF == CA[test,]$acquisition)*100;accRF
```

```
## [1] 94.66667
```

Appendix F

Logistic Regression

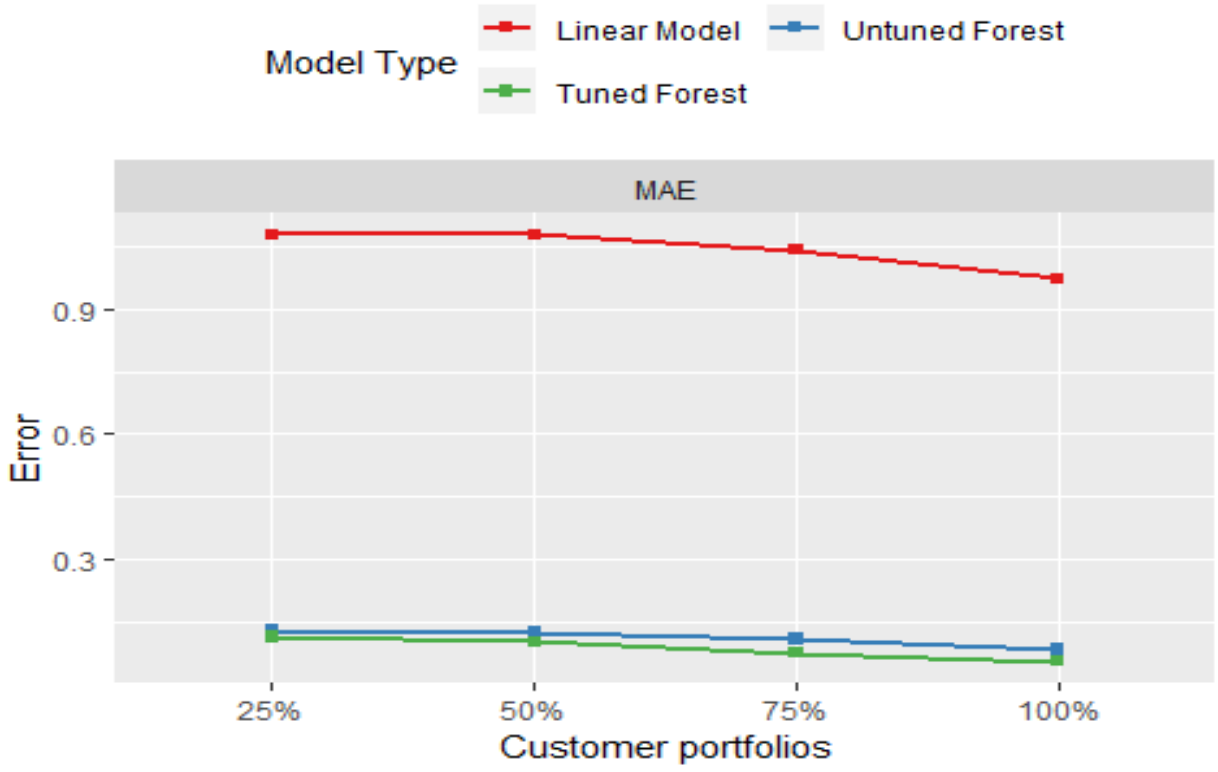
```
  log_model = glm(acquisition ~acq_expense+revenue+employees, data = CA[train
,], family=binomial)
  summary(log_model)

##
## Call:
## glm(formula = acquisition ~ acq_expense + revenue + employees,
##     family = binomial, data = CA[train, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8905  -0.1239   0.0090   0.1553   1.7430
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.592e+01  2.287e+00  -6.960 3.40e-12 ***
## acq_expense  2.547e-02  3.420e-03   7.449 9.42e-14 ***
## revenue      3.262e-02  1.445e-02   2.257    0.024 *
## employees    4.703e-03  8.081e-04   5.820 5.90e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 476.20  on 349  degrees of freedom
## Residual deviance: 135.23  on 346  degrees of freedom
## AIC: 143.23
##
## Number of Fisher Scoring iterations: 8

  predlog = predict.glm(log_model, newdata = CA[test,], type="response")
  predac = ifelse(predlog>=0.5,1,0)
  table(predac,CA[test,]$acquisition)

##
## predac  0  1
##      0 50  6
##      1 11 83

  acclog = mean(predac==CA[test,]$acquisition)*100;acclog

## [1] 88.66667
```

Appendix G

Accuracy improvements for portfolios of CLV

```
reg.linear=lm(acquisition ~acq_expense+revenue+employees+clv+industry, data =
CA[train,])

## Warning in model.response(mf, "numeric"): using type = "numeric" with a
## factor response will be ignored

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

  set.seed(123)
forestun <- rfsrc(as.factor(acquisition) ~ clv+censor+acq_expense+industry+re
venue+employees, data = customerAcquisition[train,],importance = T,
                          ntree = 1000)
set.seed(123)
forestt = rfsrc(as.factor(acquisition) ~ clv+censor+acq_expense_sq+industry+r
evenue+employees, data = customerAcquisition[train,], mtry = 4,
                           ntree = 1000,importance = T)



error.df <-
  data.frame(pred1 = predict.rfsrc(forestun,newdata = customerAcquisition[tes
t,])$predicted,
pred2 = predict.rfsrc(forestt, newdata = customerAcquisition[test,])$predicte
d,
pred3 = predict(reg.linear, newdata = CA[test,]),

             actual = customerAcquisition[test,]$acquisition,
             customer = customerAcquisition[test,]$customer)%>%
mutate_at(.funs = funs(abs.error = abs(actual - .),
                       abs.percent.error = abs(actual - .)/abs(actual)),
           .vars = vars(pred1:pred3));



#mae
error.df %>%
summarise_at(.funs = funs(mae = mean(.)),
.vars = vars(pred1_abs.error:pred3_abs.error))

##   pred1_abs.error_mae pred2_abs.error_mae pred3_abs.error_mae
## 1           0.1117316           0.0860282            1.042295

error.df2 <-
error.df %>%
left_join(CA[test,], "customer") %>%
mutate(customer_portfolio = cut(x = rev <- revenue,
breaks = qu <- quantile(rev, probs = seq(0, 1, 0.25)),
labels = names(qu)[-1],
```

```
include.lowest = T))


portfolio.mae <-
error.df2 %>%
group_by(customer_portfolio) %>%
summarise_at(.funs = funs(mae = mean(.)),
.vars = vars(pred1_abs.error:pred3_abs.error)) %>%
ungroup()

portfolio.errors <-
portfolio.mae %>%
gather(key = error_type, value = error, -customer_portfolio) %>%
mutate(error_type2 = ifelse(grepl(pattern = "mae", error_type),"MAE","MAPE"),
model_type = ifelse(grepl(pattern = "pred1", error_type),"Untuned Forest",
ifelse(grepl(pattern = "pred2", error_type),"Tuned Forest",
ifelse(grepl(pattern = "pred3", error_type),"Linear Model",
ifelse(grepl(pattern = "pred4", error_type),"Non-linear Model A",
       ifelse(grepl(pattern = "pred5", error_type),"Non-linear Model B","Non-
linear w interaction"))))),
model_type_reordered = factor(model_type, levels = c("Linear Model","Untuned
Forest","Tuned Forest")))

ggplot(portfolio.errors, aes(x = customer_portfolio,
y = error,
color = model_type_reordered,
group = model_type_reordered))+
geom_line(size = 1.02)+
geom_point(shape = 15) +
facet_wrap(~error_type2, scales = "free_y")+
scale_color_brewer(palette = "Set1") +
labs(y = "Error", x = "Customer portfolios")+

theme(legend.position = "top")+
guides(color = guide_legend(title = "Model Type", size = 4,nrow = 2,byrow = T
RUE))
```




```
error.df2 %>%
group_by(customer_portfolio) %>%
summarise(average_acquisition_expense = mean(acq_expense),
total_acquisition_expense = sum(acq_expense))
```

```
## # A tibble: 4 x 3
##   customer_portfolio average_acquisition_expense total_acquisition_expense
##   <fct>                                   <dbl>                      <dbl>
## 1 25%                                      553.                      21024.
## 2 50%                                      512.                      18956.
## 3 75%                                      561.                      20771.
## 4 100%                                     525.                      19933.
```