XXX XXXX
DA6813

Customer Acquisition Case Study

I. Executive Summary

The random forest model provided the highest prediction accuracy when forecasting customer acquisition. The logistic model and decision tree provide similar accuracy rates, however, they do not provide results quite as good as the random forest. The logistic model was fit using all meaningful interactions and non-linear functions of the variables found to be of any significance, however, the random forest not only produced better results, but it also accounted for any interaction and non-parametric terms of the variables without us needing to specify explicitly in the model like we did for the logistic.

Our random forest model produced higher accuracy rates when predicting on the upper tier customer lifetime value portfolios than it did on the portfolio of non-acquired or recently churned customers. We believe our model is highly efficient in predicting the acquisition of customers that will have lengthy duration of stay with the firm and produce high customer life time values.

II. The Problem

The task we were provided for this case study was to produce a random forest model to predict customer acquisition for the company we are working for. We then want to build a logistic model as well as a decision tree model to compare the random forest model against. The process of accurately predicting customer acquisition for our firm will allow us to identify the most prominent predictors or classifier of our response and allow us to administer a higher amount of firm resources into customers potential customers fitting those qualifications. Additionally, we desire to build portfolios dependent on a customer's specific lifetime value and identify which echelons we can most accurately predict. Ideally, we would like to be able to most accurately predict the highest clv customers and confirm the accuracy of the model with the firms top and most profitable customers.

In summary, the main purposes of this study are to identify the best predictive model, most significant variables and their effect on customer acquisition, and lastly test the optimal model against different portfolios of customers. The remainder of the report will discuss a brief introduction into literature related to the processes utilized in our study, followed by a discussion of the methodology taken in our procedures, the steps required to properly process the dataset, and then we will end with a discussion of the findings of our analysis.

III. Review of Related Literature

Random forests are a scheme proposed by Leo Breiman in the 2000's for building a predictor ensemble with a set of decision trees that grow in randomly selected subspaces of data. According to a study in the Journal of Machine Learning Research by Gerard Biau, despite growing interest and practical use, there has been little exploration of the statistical properties of random forests, and little is known about the mathematical

forces driving the algorithm. The random forest is still a relatively new machine learning method, however, the amount of literature testing these mathematical forces and statistical properties behind the model is largely increasing. Studies are being recommended to test the algorithm and method behind the way the model collects its out of bag error. This is a largely unstudied parameter of the model. The method of random forest may be relatively new, but the results and accuracy it produces can not be denied. The future of the random forest will surely provide even greater methods to enhance prediction power and provide a more concrete understanding of the underlying techniques used for its creation.

IV. **Methodology**

In this case study we will be comparing the results of a random forest model, decision tree model, and lastly a logistic model. For the random forest model we will create a hyper grid of the parameters mtry(number of variables tested per node) and ntree(number of trees in the RF model), as well as nnodes(or number of nodes) and use this hyper grid to loop over the random forest model and identify the most optimal set of parameters to use for our model. We will also prune the decision tree model to provides the most accurate tree possible. Lastly, we will use variable interaction plots and functions to identify possible significant interaction variables in our model as well as use a general additive model to identify any potential non-linear aspects we need to account for in our model. These interactions and non-parametric terms will be added in the logistic model appropriately to test against the other methods discussed above.

The data will be split using a 70/30 ration from training to test set. The training data is comprised of 350 observations and the four main variables of analysis including acq_expense(amount of money used in marketing efforts), industry(B2B or other), revenue(of potential customer firm), and employees(of potential customer firm). We will also be considering the interaction of these terms as well as non-parametric squared, cubic, or quartic functions of the variables. The test set will be comprised of the same variables, however, it will only have 150 observations.

Additionally, a variable of customer lifetime value(clv) will also be used to build portfolios of customers (non-acquired or recently churned, medium lifetime value, and high lifetime value).

The logistic model operates under several fundamental assumptions that contribute to the accuracy of the results of the model. These assumptions are as follows: the predictor must have a linear relationship with the log odds of the response variable, the variables must have no intercorrelations or multicollinearity, there are no major outliers in its continuous predictors, normality of the data, as well as homoscedasticity. Failure of the data to abide by these assumptions can lead to inaccurate or false results and thus all these assumptions will be tested in our analysis.

The decision tree is a non-parametric test that has no assumptions. It simply splits the data into nodes that can be used to classify and predict on the data. The nodes are determined by which split in the data or variable would reduce the residual sum of squares the most and thus the split is made at this point.

The random forest is a culmination of a large amount of decision trees created using bootstrapping random sampling techniques to predict upon a response variable. The trees build in the random forest model are uncorrelated and use only a subset of predictors to choose from at each node and tree.

These are the three main methods of analysis we will be using as well as a general additive model to create smoothing splines of our continuous variables and test the linearity of them. We will also be using interaction functions and importance plots to identify the optimal parameters, interactions, and variables in question.

V. Data

The data was relatively clean to start out with. The customer acquisition data set was provided to us with 500 and 16 variables. The data did not have any missing values and thus there was no need for imputation of the data. The variables acquisition, industry, and censor were all being treated as numeric variables when in the initial data, however, we converted the three variables into factors due to their binary nature. While missing data or variable cleaning was not a large issue in the data, there was an issue with several variables and their correlation with the response variable, acquisition.

The main problem with a large amount of the variables was that in order for a customer to have information or a value it was indicative of whether or not the customer had been acquired. An example of one of these variables is first_purchase, for a customer to have made a first purchase it is a requirement that the company would have acquired them as a customer. The issue here is that the variables were perfectly predicting our response variable.

The result of this finding, we must not include these variables in our modelling process. The data dictionary provided adequate definitions of the variables we needed to identify and remove, however we wanted concrete evidence to affirm our removal process. A table was created for every variable using the predictor and the response variable, a picture has been provided below for reference.

```
head(table(CA$first_purchase,CA$acquisition))
```

```
##
##                      0   1
##   0                208   0
##   3.18885119408626   0   1
##   5.39872219587692   0   1
```

As we can see first_purchase was perfectly separated with acquisition and thus required removal. In total, 9 variables were identified and chosen to remove from the model based on the previously mentioned reason. Those variables were first_purchase, clv, duration,ret_expense, ret_expense_sq, crossbuy, frequency, and frequency_sq.

After removing 9 variables from the models we build this left us with only five variables that could be considered for use, as we did not feel that customer number was appropriate to include. This is a large limitation of the dataset, we not only had a small

number of variables to build the models using but we also had a small amount of data to build the models with, especially when divided into a test and a training subset.

VI. Findings (Results)

Our analysis showed that the random forest model had the highest accuracy rate of all the models tested. It produced an accuracy rate of 89.3%. The hyper parameters for this model was 1000 trees and an mtry value of 3. The logistic model was the next highest, with the decision tree being the lowest ranking model of comparison.

```
## [1] "Log Model:"        "0.886666666666667"

## [1] "Log Model with Interaction:" "0.866666666666667"

## [1] "Decision Tree:" "0.88"

## [1] "Optimal Random Forest:" "0.893333333333333"
```

The first step of analysis was to determine if the variables followed the assumptions of the logistic regression. All three continuous variables appear to be normal in distribution or very close to it. The variance inflation values of the logistic model show no issues with multicollinearity amongst our predictors as well. The base assumptions of the model appear to have been met.

In our analysis we used a GAM model to test the non-parametric terms of the predictors. The acq_expense squared terms was a variable already provided in the dataset, however, we found that it did not test significantly in the non-parametric ANOVA of the GAM model. Additionally, we were unable to identify any significant non-parametric variables for our model analysis. We even tested the acq_expense variable in our original logistic model. It did not provide any increase or decrease to the accuracy readings of the model.

Interaction terms were also tested using the find.interaction function in R. The interactions between acq_expense and employees had the highest difference between paired and additive estimations with acq_expense and revenue having the next highest. We decided to test the significance of these two interactions by creating a separate logistic model with interaction included. The accuracy rate of the model decreased by 2% when both interaction terms were included. The interaction of acq_expense employees produced a p-value 0f 0.086 and is very close to the commonly used cutoff of 0.05, however the interaction of acq_expense and revenue did not test near a significant value. We decided to keep the interaction of acq_expense and employees in the model and test the accuracy rate again. The accuracy rate of the model stayed the exact same when just this interaction was included.

The addition of interaction terms and non-parametric terms into the model did produce terms that tested statistically significant. Both acq_expense_sq and the interaction between acq_expense and employees were significant in the logistic model, however, we did not see a rise in the accuracy of the model because of this. The random forest model it seems can account for all of these discrepancies without the need for our encoding and accounting for.

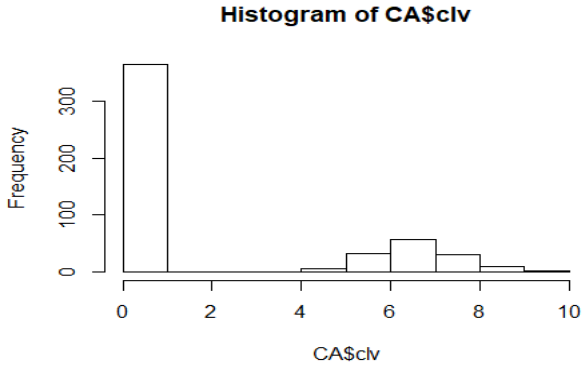
The next issue we wanted to determine was variable significance. In the random forest, the most significant predictor was acq_expense, followed by employees.

If we summarize the results of the logistic model, we can see similar interpretations of the variable importance. In fact, the p-values of the logistic model perfectly agree with the importance plot of the random forest. Acq_expense produced the lowest p-value, followed by employees, revenue, and lastly industry was the sole non-significant predictor of acquisition. The estimation coefficients of all variables were rather small, however, we can see that as acq_expense, revenue, and employees all increase the chance of a customer being acquired also increase.

```
                                ## Call:
## glm(formula = acquisition ~ acq_expense + industry + revenue +
      ##     employees, family = "binomial", data = CAtrain)
                                ##
                    ## Deviance Residuals:
          ##     Min       1Q   Median       3Q      Max
          ## -3.7936  -0.1325   0.0100   0.1442   1.6824
                                ##
                      ## Coefficients:
   ##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.618e+01  2.299e+00  -7.041 1.91e-12 ***
## acq_expense  2.533e-02  3.409e-03   7.431 1.07e-13 ***
   ## industry1    5.585e-01  4.658e-01   1.199   0.2305
  ## revenue      3.309e-02  1.440e-02   2.298   0.0215 *
## employees    4.619e-03  8.053e-04   5.735 9.72e-09 ***
```

Lastly, we constructed three portfolios based on the customer lifetime value variables provided in the dataset. If we look at the distribution of customer lifetime value we can see that about half of the observations have a value of 0, indicating either non-

acquisition or a customer that has churned and left the company. The distribution then jumps to a value of around 4 and increases to around 10. A histogram has been provided below. The customers with non-zero CLV's have a mean of 6.5 as their value.



The three portfolios of customers are as follows:

1. Non-Acquired or Churned, CLV = 0
2. Medium CLV Customers, 6.5 =< CLV > 0
3. High CLV Customers, CLV > 6.5

We used these three portfolios to see which areas or portfolios customers our models were most accurate on. The results are provided below.

	Logistic	Interaction	Decision Tree	RF
High CLV:	8.00	12.00	8.00	8.00
Medium CLV:	11.11	16.67	10.04	11.11
Not Acquired or Churned:	12.15	13.08	14.02	11.21

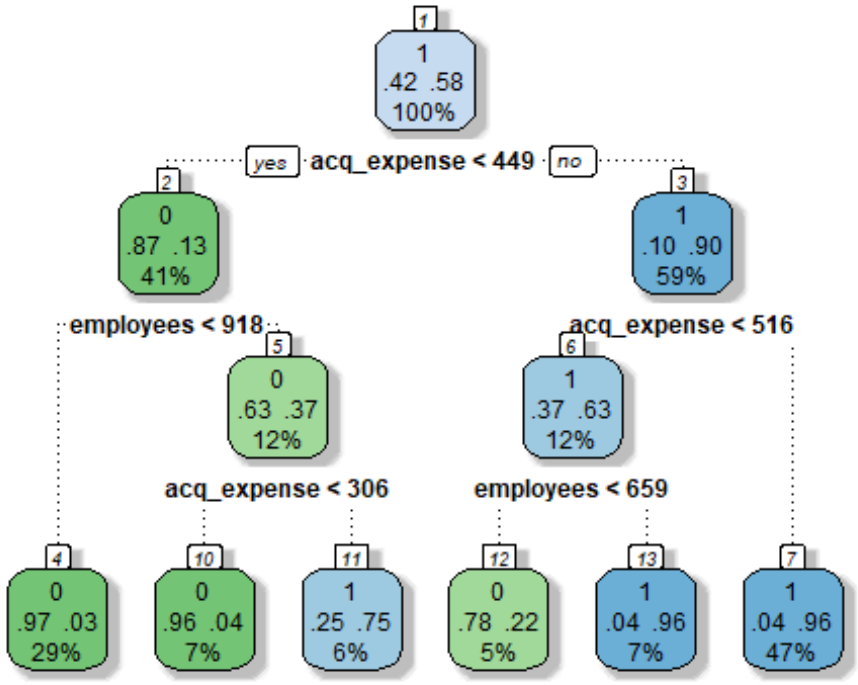
As we can see our models on average were the best at prediction High CLV customers and as were the least accurate at predicting the non-acquired or churned customers. To me this makes a bit of sense as a customer in this portfolio can be customer who were never acquired or customer who were acquired but decided to leave the firm. However, being able to more accurately predict the upper echelon customers is a large bonus as those are the customers providing the most profitability and good business for the firm.

VII. Conclusions and Recommendations

In conclusion, we recommend using the random forest model for prediction of customer acquisition. It is the most accurate model and it accounts for interaction and non-linearity of the variables in its methodology, versus the logistic model in which we need to account for them manually which is not always easy to do. Acq_expense and employees were the most important and significant predictors of the response variable, acquisition.

I believe it would also benefit our study if we were to analyze what variables and model have the greatest prediction power of how long a customer stays with the company. The ability to identify and accurately predict the acquisition of a customer as well as the lifetime of the that customer would be of great value to a company. In doing this it would allow the use of many of the variables that we were required to cut from the model of acquisition. The random forest model is a relatively new methodology and will surely see many improvements and new potential uses in the future. As the random forest evolves and new studies are released so too should the model we have created.

Appendix:




```
log.model = glm(acquisition ~ acq_expense +
                  industry + revenue + employees, data = CAtrain, family = "b
inomial")
log.preds = predict(log.model, newdata = CAtest, type = "response")
log.pred.class = ifelse(log.preds >= 0.5, 1, 0)
table(as.factor(log.pred.class),y)

##    y
##      0  1
##   0 50  6
##   1 11 83

log.acc = 1 - mean(log.pred.class!=y);log.acc
```

```
## [1] 0.8866667
```

```
summary(log.model)
```

```
##
## Call:
## glm(formula = acquisition ~ acq_expense + industry + revenue +
##     employees, family = "binomial", data = CAtrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7936  -0.1325   0.0100   0.1442   1.6824
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.618e+01  2.299e+00  -7.041 1.91e-12 ***
## acq_expense  2.533e-02  3.409e-03   7.431 1.07e-13 ***
## industry1    5.585e-01  4.658e-01   1.199   0.2305
## revenue      3.309e-02  1.440e-02   2.298   0.0215 *
## employees    4.619e-03  8.053e-04   5.735 9.72e-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: 133.78  on 345  degrees of freedom
## AIC: 143.78
##
## Number of Fisher Scoring iterations: 8
```

```
#Build the optimal random forest model
set.seed(123)
rf.opt.model = randomForest(acquisition ~ acq_expense + industry +
                            revenue + employees,
                            data = CAtrain, mtry = opt_mtry, ntree = opt_ntre
e, importance = TRUE)
rf.opt.preds = predict(rf.opt.model, newdata = CAtest, type = "class")
rf.opt.model2 = rfsrc(acquisition ~ acq_expense + industry +
                            revenue + employees,
                            data = CAtrain, mtry = opt_mtry, ntree = opt_ntre
e, importance = TRUE)
rf.opt.preds2 = predict.rfsrc(rf.opt.model2, newdata = CAtest)
table(rf.opt.preds, y)
```

```
##             y
## rf.opt.preds  0  1
##            0 51  6
##            1 10 83
```

```
rf.opt.acc = 1 - mean(rf.opt.preds!=y);rf.opt.acc
```

```
## [1] 0.8933333
```

```
# cross-check with vimp
find.interaction(rf.opt.model2,
                      method = "vimp",
                      importance = "permute")
```

```
## Pairing acq_expense with employees
## Pairing acq_expense with revenue
## Pairing acq_expense with industry
## Pairing employees with revenue
## Pairing employees with industry
## Pairing revenue with industry
##
##                                Method: vimp
##                     No. of variables: 4
##            Variables sorted by VIMP?: TRUE
##    No. of variables used for pairing: 4
##     Total no. of paired interactions: 6
##              Monte Carlo replications: 1
##     Type of noising up used for VIMP: permute
##
##                          Var 1   Var 2 Paired Additive Difference
## acq_expense:employees 0.3369  0.0628 0.3625   0.3997    -0.0372
## acq_expense:revenue   0.3369  0.0005 0.3355   0.3374    -0.0020
## acq_expense:industry  0.3369 -0.0011 0.3368   0.3358     0.0009
## employees:revenue     0.0637  0.0005 0.0630   0.0642    -0.0012
## employees:industry    0.0637 -0.0011 0.0638   0.0626     0.0012
## revenue:industry      0.0009 -0.0011 0.0004  -0.0002     0.0006
```

```
log.model.interaction = glm(acquisition ~ acq_expense +
                  industry + revenue + employees + acq_expense*employees
                  + acq_expense*revenue, data = CAtrain, family = "binomial")
log.preds.interaction = predict(log.model.interaction, newdata = CAtest, type
= "response")
log.pred.class.interaction = ifelse(log.preds.interaction >= 0.5, 1, 0)
table(as.factor(log.pred.class.interaction),y)
```

```
##    y
##      0  1
##   0 49  8
##   1 12 81
```

```
log.acc.interaction = 1 - mean(log.pred.class.interaction!=y);log.acc.interac
tion
```

```
## [1] 0.8666667
```

```
NewCLV <- CA$clv
NewCLV <- subset(NewCLV, NewCLV > 0)
summary(NewCLV)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.272   5.907   6.543   6.579   7.128   9.225
```

```
error.df <-
  data.frame(pred1 = as.factor(log.pred.class),
             pred2 = as.factor(log.pred.class.interaction),
             pred3 = dt.preds,
             pred4 = rf.opt.preds,
             actual = CAtest$acquisition,
             customer = CAtest$customer)
quantile(CAtest$clv)
```

```
##       0%      25%      50%      75%     100%
## 0.000000 0.000000 0.000000 5.386675 8.732700
```

```
error.df$Portfolio <- ifelse(CAtest$clv <= 0.0, "NA or Churned",
                             ifelse(CAtest$clv <= 6.5, "Medium CLV", "Highest
CLV"))
error.df$clv = CAtest$clv
error.df$Portfolio = as.factor(error.df$Portfolio)
summary(error.df)
```

```
##  pred1  pred2  pred3  pred4  actual    customer             Portfolio
##  0:56   0:57   0:51   0:57   0:61   Min.   :  3.0   Highest CLV  : 25
##  1:94   1:93   1:99   1:93   1:89   1st Qu.:126.5   Medium CLV   : 18
##                                     Median :265.5   NA or Churned:107
##                                     Mean   :262.4
##                                     3rd Qu.:398.8
##                                     Max.   :499.0
##       clv
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :1.858
##  3rd Qu.:5.387
##  Max.   :8.733
```

```
names(error.df) <- c("Logistic", "Logistic Interaction", "Decision Tree", "Op
timal Random Forest",
                     "Actual", "Customer", "Portfolio", "CLV")
```

Create a data frame for each portfolio

```
Portfolios <- split(error.df,error.df$Portfolio)
HighCLV <- Portfolios[[1]]
```

```
MediumCLV <- Portfolios[[2]]
DidNotAcquire.or.Churned <- Portfolios[[3]]

HighCLV <- HighCLV[,c(1,2,3,4,5)]
MediumCLV <- MediumCLV[,c(1,2,3,4,5)]
DidNotAcquire.or.Churned <- DidNotAcquire.or.Churned[,c(1,2,3,4,5)]
```

Calculate the error rate for each model for each portfolio

```
HighCLV.err = HighCLV[,-c(1:5)]
MediumCLV.err = MediumCLV[,-c(1:5)]
DidNotAcquire.or.Churned.err = DidNotAcquire.or.Churned[,-c(1:5)]
for (i in 1:4) {
  HighCLV.err[,i] = mean(HighCLV[,i]!=HighCLV$Actual)
  MediumCLV.err[,i] = mean(MediumCLV[,i]!=MediumCLV$Actual)
  DidNotAcquire.or.Churned.err[,i] =
    mean(DidNotAcquire.or.Churned[,i]!=DidNotAcquire.or.Churned$Actual)
}
HighCLV.err <- HighCLV.err[1,]
MediumCLV.err <- MediumCLV.err[1,]
DidNotAcquire.or.Churned.err <- DidNotAcquire.or.Churned.err[1,]
```

Create a data frame of all the model errors for each portfolio

```
Model.Errors.Portfolio <- rbind(HighCLV.err, MediumCLV.err, DidNotAcquire.or.
Churned.err)
names(Model.Errors.Portfolio) <- names(HighCLV[,-5])
rownames(Model.Errors.Portfolio) <- c("High CLV", "Medium CLV", "Not Acquired
or Churned")
Model.Errors.Portfolio <- round(Model.Errors.Portfolio*100,2)
```

As we can see on average we were able to predict the higher CLV(above 6.5) value customers with the highest accuracy. The customer who were not acquired or had previouslt churned from the company was the group we had the highest average error predicting.

```
Model.Errors.Portfolio
```

```
##                          Logistic Logistic Interaction Decision Tree
## High CLV                     8.00                12.00          8.00
## Medium CLV                  11.11                16.67          5.56
## Not Acquired or Churned     12.15                13.08         14.02
##                          Optimal Random Forest
## High CLV                                  8.00
## Medium CLV                               11.11
## Not Acquired or Churned                  11.21
```