CHURN PROFILING AND PREDICTIVE MODELLING

Acaso
Analytics

ACTL3142

Executive Summary

Overview:

This report is an investigation into the Churning of customers for TelecomUSA, where predictions for Churn and Churn profiling will be provided. The basis of this analysis is the information on 40000 selected customers and predictions will be made on 10000 new customers of interest.

Methodology:

To find the optimal predictive model the data must first be explored and cleaned. This data is then used to fit multiple different statistical learning models which are evaluated on multiple criteria that give indication for their performance on the unseen data. This modelling will allow for prediction of probability of churn on new customer data points, as well as give insight into what factors contribute to a customer's churn.

Solution:

The optimal predictive model is the gradient boosted tree model, which had a 43.05% precision when predicting 3000 customers as 'Yes'. This is a 49.2% increase over random guessing, which would result in 28.8%. The most influential predictors for an individual Churn can be identified through this model as well, with the 5 most important predictors for Churn being CurrentEquipmentDays, MonthsInService, MonthlyMinutes, MonthlyRevenue, and CreditRating.

Data Exploration, Cleansing and Initial Analysis

Data exploration is the first step to any modelling activity. It gives initial insight into the variables and ensures that the data valid. If there are any issues with the way the data is collected or presented, the data is cleaned and prepared in a way that is easy to understand by the statistical learning models. Extra modifications such as creation of new variables and train/test splits can also be done in this stage.

Data Inspection and Cleansing:

TelecomUSA has provided two datasets, both containing a range of customer attributes for each data point. The first dataset (40000 entries) will be used to train and validate the statistical learning models as the outcome of interest, customer churn, is included. This dataset will also be used to identify customer attributes that may indicate a higher probability of churn. The second dataset (10000 entries) will be used for prediction of future customer churn.

After initial inspection of the data, there are many data points that contain missing values. These values are coded as either 'NA', 'Unknown' or ''. As these issues are prevalent in large quantities in both datasets, removing all these data points are unfavourable. Therefore, data imputation (replacement of missing data) is performed on these data points, see Appendix 1.1 for more details and a list of impacted variables. The assumption is made that these values are missing at random (MAR) and therefore depend on the other observed values within a datapoint. This assumption can be justified due to the wealth of data provided and the many predictors present that can inform these variables.

Another alternation made to the dataset is the removal of the ServiceArea predictor. This predictor has 718 levels with many having extremely low numbers of entries, which complicates the models for little gain. It is assumed that the information from this predictor is not worth keeping in the data set as there are other predictors, such as PrizmCode and DroppedCalls, that contain the relevant information in a more concise manner.

A train, 30000 data points, and test, 10000 data points, split is also created which will be used for creating confusion matrices and plotting some graphical examples.

The proportion of the data is also inspected, and it is found that ~28.8% of the data relates to the 'Yes' class and ~71.2% for the 'No' class. These proportions may be problematic, therefore a train set for up and down sampled data will also be created in case they are the optimal sampling method after testing.

Data Exploration and Analysis:

To get some initial insights into the data, general data exploration is done. As the variable of interest is Churn a good starting point is the correlation between itself and the predictors. While this can give a good indication of relationships within the data, is limited due to correlation only showing linear relationships. This can also guide our graphical interpretations to variables that are less likely to be random noise. Figure 1 shows the top 10 highest absolute value correlation metrics, where the highest is a correlation of 0.1. See Appendix 1.2 for a full list of correlations.

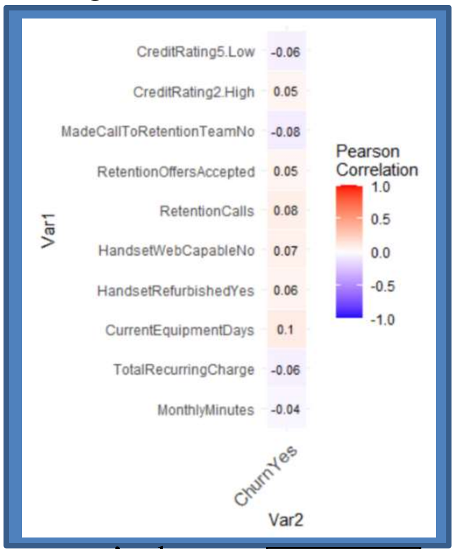


Figure 1

These correlations give indication to variables that may be predictive of a customer's churn. We can further investigate the high correlation variables through density plots (for continuous variables) and stacked bar plots (for categorical variables). Shown in Figure 2 is the density plot, relative to churn, for the continuous variable CurrentEquipmentDays (0.1 correlation) and HandsetPrice (0 correlation). The dashed lines indicate the relative average for churn customers vs non-churn. Through the use of these graphs it is obvious that the correlated variable has distinct distributions, and hence averages, for churn vs non-churn, while little difference is seen in the non-correlated variable.

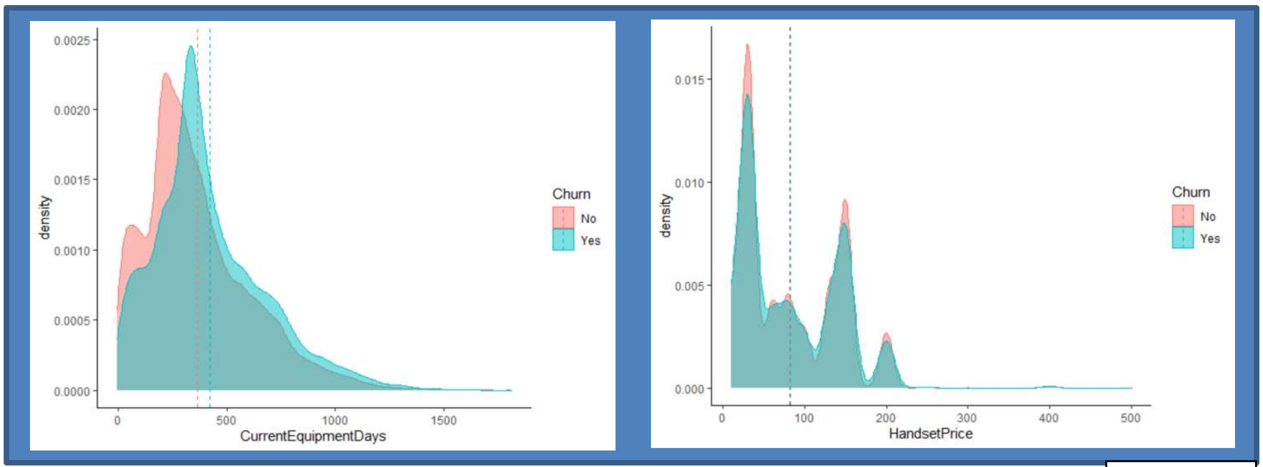


Figure 2

Shown in Figure 3 is the stacked bar plot, relative to churn, for the categorical variable MadeCallToRetentionTeam (0.08 correlation) and OwnsComputer (0.01 correlation). These plots show the relative proportion of churn vs non-churn for each category, with the dashed line representing the average proportion split between churn (28.8%) and non-churn (71.2%). Through the use of these graphs it is obvious that the correlated variable does deviate heavily from the average split, while the uncorrelated is approximately equal to it.

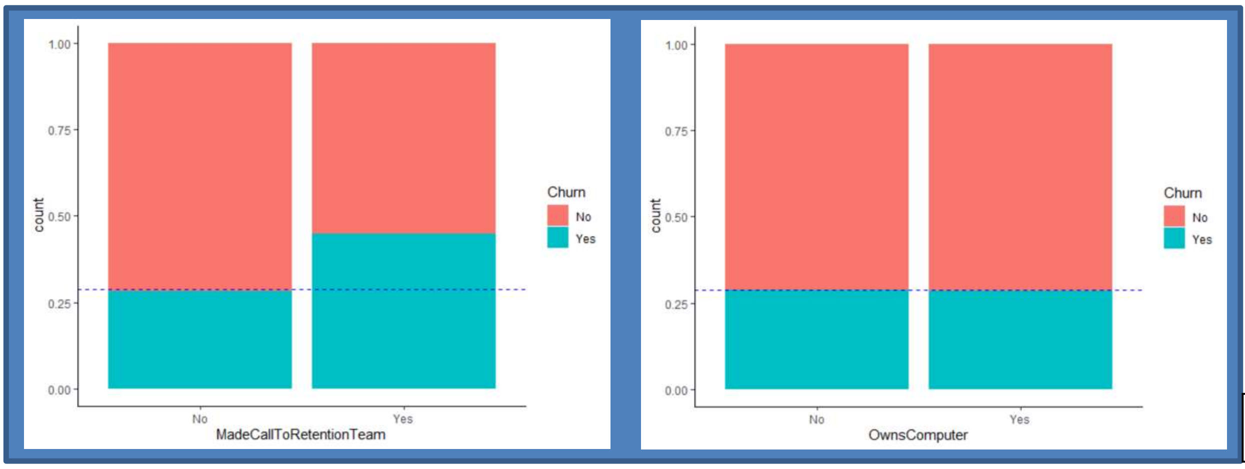


Figure 3

See Appendix 1.3 for the plots of the other top correlated variables, as well as code that can be reproduced to gain access to each variable's plot.

Choice of Optimal Predictive Model

There are many statistical learning models available to use, but the optimal model is data dependent. Therefore, it is essential to test many different types of model before deciding on which to use. As the goal is to find the 3000 customers most likely to churn, the main metric to be used to tune and ultimately choose the optimal model will be the area under the Precision-Recall Curve (AUC_RP). This metric will be supported by the precision and recall/specificity when predicting 3000 customers as 'Yes', as well as the more general area under the ROC curve (AUC_ROC). For further information and justification on these metrics, see Appendix 2.1.

Models Considered:

To find the AUC_RP for each method, and consequently the choice for the optimal method, 5-fold cross validation will be performed for each model. This will ensure that the resulting AUC is valid and can be confidently used for comparison. One example of the resulting RP curve will also be plotted and reported. Similar will be done for the AUC_ROC.

The precision and specificity will be found by applying a probability threshold that results in 3000 'Yes' predictions and averaging the relative metrics over a 5-fold cross validation, see Appendix 2.2 for code. One example confusion matrix for the threshold will also be reported.

If hyperparameter tuning is required, a grid search will be done with an extensive combination of relevant hyperparameters. 5-fold cross validation will be performed on each combination and the combination with the highest AUC_RP will be chosen for the final model. For more information and the set-up for cross-validation and grid search, see Appendix 2.3.

To address the possible issues with the unbalanced data set, a set of cross validation will be performed for the raw data, up-sampled data and down-sampled data and the highest AUC_RP and relative sampling method will be reported.

A range of models were considered through the aforementioned analysis. The results, as well as a brief description and analysis of each model, can be found in Appendix 2.4. For a concise summary and comparison of all models see Appendix 2.5.

Chosen Model:

Through testing a wide variety of models it was found the weakest models were the linear models and the better performers were those with non-linear decision boundaries or those that captured non-linear trends. This indicates that most of the variables follow non-linear trends and possibly have interactions, which is supported by the low correlations between the variables and the response variable Churn. No sampling was also the best way to sample the data when maximising the AUC_RP 9/10 times, which holds true for the boosted trees as well.

The best performing model was the gradient boosted trees, with 250 trees, 4 depth and 0.01 shrinkage, dominating all other methods in each metric. The metrics are given below. To compare it to other methods, refer to Appendix 2.5.

AUC_RP	AUC_ROC	Precision	Sensitivity	Sampling
0.4326199	0.6617765	0.4305	0.4539	None

Using the precision score of 0.4305, we can expect that in the evaluation data out of the 3000 'Yes' predictions 43.05% are true. This is a 49.4% improvement over random classification, which would result in an approximate success rate of 28.8%.

The AUC_RP is a 50.2% improvement over the baseline (random classifier) measure of 0.288. A higher AUC_RP describes the average performance of the classifier in terms of Precision and Sensitivity.

The AUC_ROC is a 32.35% improvement over the baseline (random classifier) measure of 0.5. A higher AUC_ROC describes the average performance of the classifier in terms of Sensitivity and Specificity.

Plotted in figure 4 are the PR and ROC curves of all other methods compared to the gradient boosted tree model. For larger plots with each model detailed refer to Appendix 2.5.

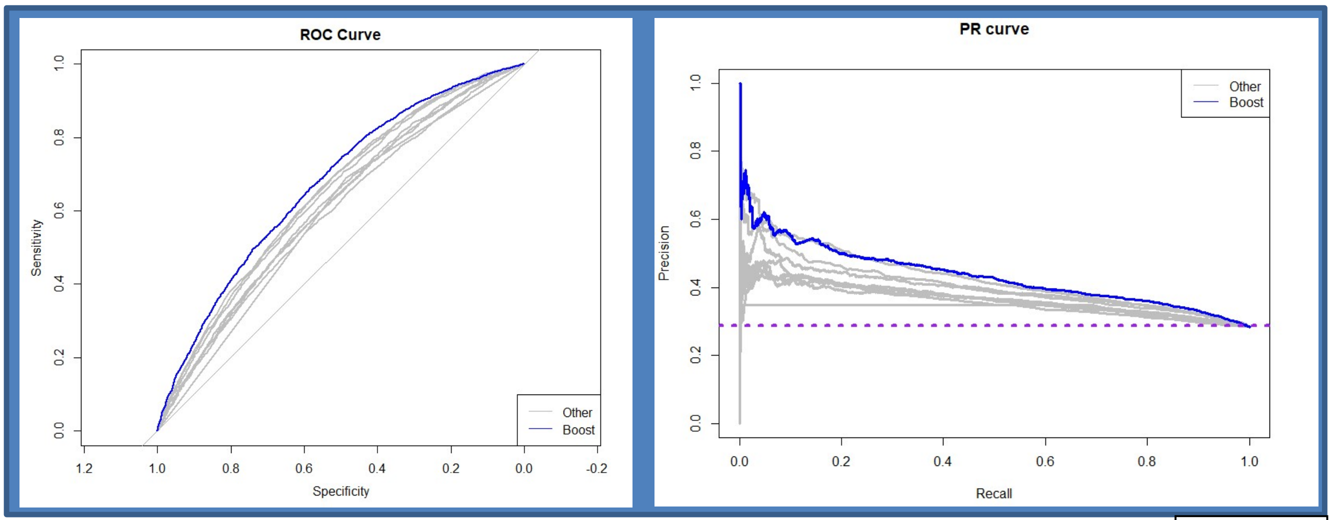


Figure 4

While the decision-making process within the boosted model isn't as easily interpretable as a simple decision tree or linear model, the drastic increase in predictive power is a worthy trade-off. Despite this, the boosted model still has lots of tools for interpretation and analysis, such as relative influence and marginal predictor plots.

According to the relative influence diagnostics of the boosted model we can determine that the 5 most important variables to classifying whether a customer will churn are CurrentEquipmentDays (1612 rel. inf), MonthsInService (12.83), MonthlyMinutes (8.05), MonthlyRevenue (5.47) and CreditRating (5.3). For an exhaustive list of relative influences see Appendix 2.6.

The marginal predictor plots exhibit how the probability of churn is impacted by each predictor irrespective all other predictors. As the data is not resampled, we can compare these plots to the baseline probability of churn of 28.8% to see when an individual is more/less likely than average to Churn. Further analysis of these plots will be utilised in the next section to create a customer churn profile.

Customer Churn Profile

A proactive churn management system can be created based on the results of the gradient boosted model. The relative influences and marginal predictor plots can be inspected to get an idea of what predictors are most important for classification of customer Churn, in addition to when a customer is more likely than average (>28.8%) to churn based on a specific predictor. Analysis was conducted on the most interesting influential predictors and some summaries were made based on groups of predictors that are influential in this model and others.

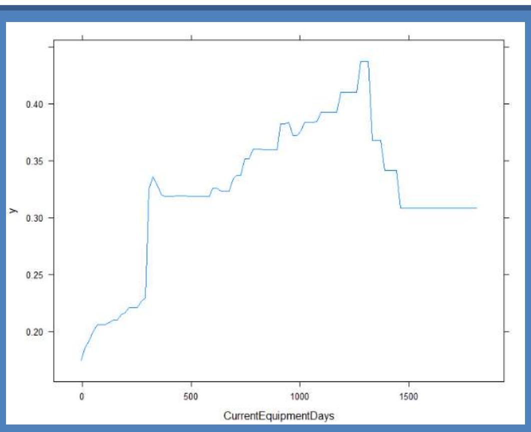


Figure 5

<u>CurrentEquipmentDays:</u>

At 0 days the probability of Churn is ~18%. This increases slowly overtime until the 300 days. At the ~300 days mark there is a spike up to ~31% likelihood of churn. This is assumed to be due to the end of a payment plan. This is the level likelihood until 600 days where it begins to increase again linearly. This is plotted in figure 5.

Proactive measures can be put in place to ensure an individual is enticed to upgrade their device and therefore renew their service with the company, particularly towards and after the plan end.

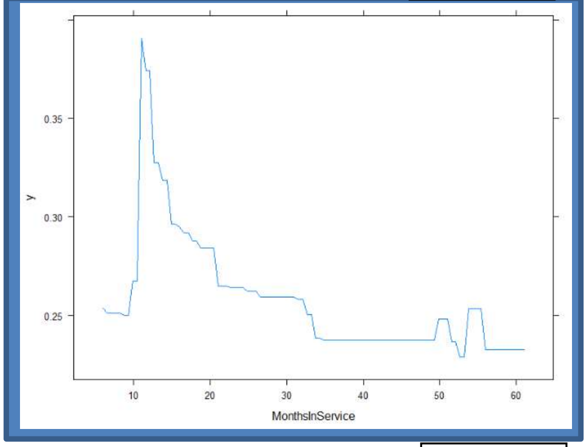


Figure 6

<u>MonthsInService:</u>

At 0 months the probability of Churn is ~25%. This stays level until 11 months where the likelihood spikes to ~40%. This is assumed to be due to the end of plan. However, at month 12 there is a sharp decline in Churn probability, quickly falling below average at ~15, and the likelihood continues to decrease beyond this. This is plotted in figure 6.

Proactive measures can be put in place to ensure an individual rolls their plan over with the company. The greatest marketing efforts should be placed on those finishing their first plan.

MonthlyMinutes:

At 0 minutes the likelihood to churn is approximately 60%. This likelihood decreases exponentially, levelling out at ~26% at 350 minutes. Therefore, advertising should be focused at users with <350 minutes, with focus increasing as minutes decrease.

MonthlyRevenue:

The larger amount of monthly revenue the more likely to churn, growing from ~20% at $0 to ~35% at ~$240. This is likely due to charges from roaming calls or overage minutes in combination to the normal plan pricing. Deals should be offered to those who produce revenue >$60 as this is the point where they become more likely to churn than average, with better deals offered to those who produce more revenue.

TotalRecurringCharge:

Churn spikes to ~30% at ~$10 and ~$30 recurring charge, and decreases linearly down to ~22% at ~$100. Plans at these prices may be considered lower value than others in the market and should be investigated.

CreditRating:

Medium-Lowest credit ratings are slightly less likely to churn (~24-26%), while Good-High are slightly more likely to churn (~29%). Focus advertising to higher credit rating customers as they are reliable income.

Relevant marginal predictor plots can be found in Appendix 3.1 and a customer churn infographic with extra relevant predictors with general analysis and advice can be found attached in Appendix 3.2.

Limitations

While this analysis is extensive, there are still some limitations that need to be noted:

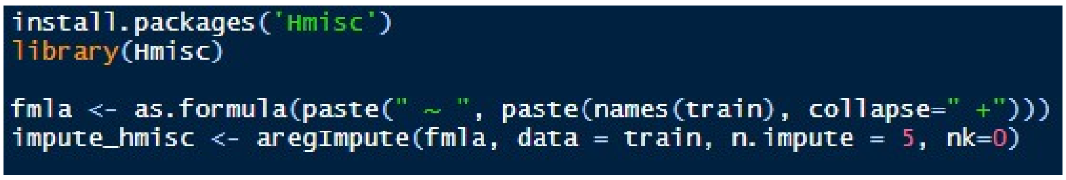
- The models, hyperparameters and sampling type are tuned based on this exact scenario where 3000 customers will be predicted as ‘Yes’. If a better general classifier was required a better tuning method would likely be the AUC_ROC as it considers both classes rather than the single positive class.
- Increasing the amount of ‘Yes’ predictions results in declining precision, and this impacts the quality of the prediction when the model is forced to predict 3000 customers as ‘Yes’.
- The best precision acquired was 0.4305. While this is a great improvement over the random model (~49% improvement from 0.288), it still leaves a lot to be desired and revenue may be lost when offering deals or increasing advertising to those who wouldn’t churn either way. This trade-off would need to be investigated further to determine what precision is suitable to ensure has a net positive impact.
- More predictors that closely relate to churn could be attained through methods such as surveying for satisfaction with the service and reasons why a churn occurred.
- While a wide variety of models were considered, there are still a wealth of possible models, such as Neural networks, to use, and therefore the true optimal model may not have been found.

APPENDIX

1 – Data Exploration, Cleaning and Initial Analysis

1.1 – Imputation

Imputation of data is done using the ‘Hmisc’ package on R, which allows us to utilise the non-parametric approach of predictive mean matching to predict the missing values based on other values for a given data point.

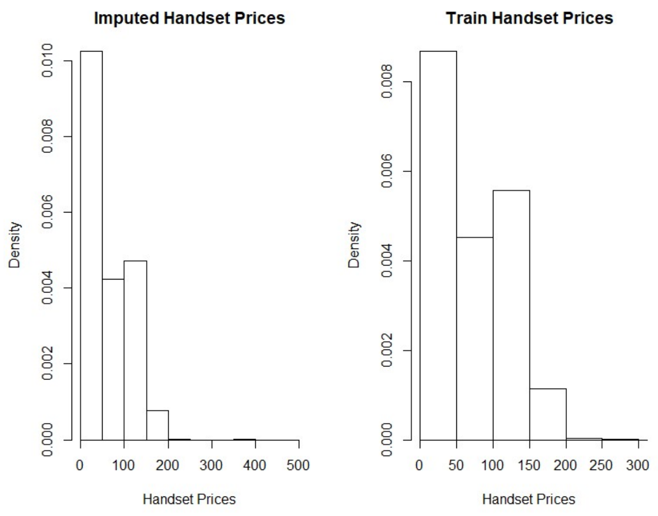


```r
install.packages('Hmisc')
library(Hmisc)

fmla <- as.formula(paste(" ~ ", paste(names(train), collapse=" +")))
impute_hmisc <- aregImpute(fmla, data = train, n.impute = 5, nk=0)
```

This imputation is done 5 times (due to the inherent randomness in the method). For continuous values the average value of these 5 imputations is taken. For the categorical variable a similar approach is taken, where the majority vote is taken.

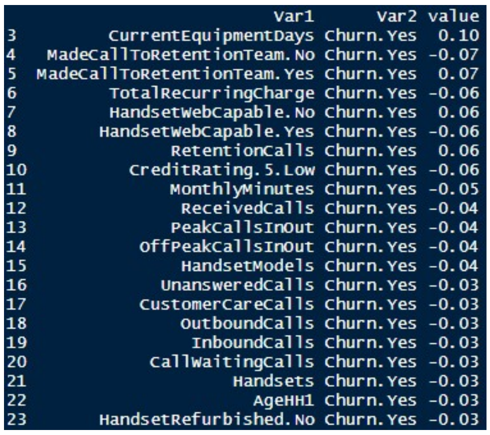
The impacted variables and their relative amount of missing values in the training set are: MonthlyRevenue (0.31%), MonthlyMinutes (0.31%), TotalRecurringCharge (0.31%), OverageMinutes (0.31%), RoamingCalls (0.31%), Handsets (<0.01%), HandsetModels (<0.01%), CurrentEquipmentDays (<0.01%), AgeHH1 (1.79%), AgeHH2 (1.79%), HandsetPrice (56.87%) , MaritalStatus (38.52%).

This method seems to be an effective way to impute these values, as we can see by comparing the density between Handset Prices in both the training set and imputed set:

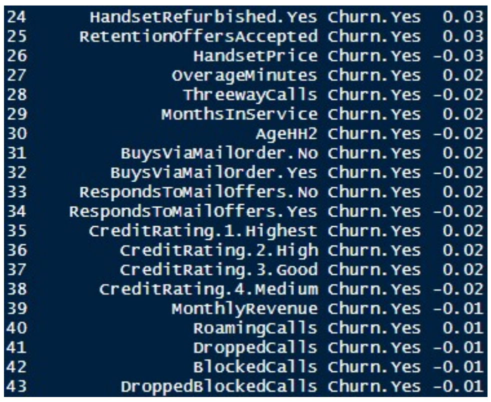


1.2 - Correlation

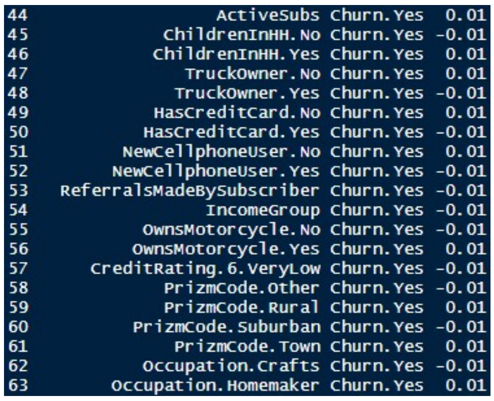
Full Correlation List:



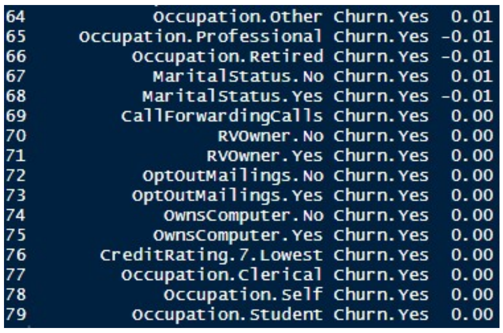
```
                      Var1      Var2 value
3        CurrentEquipmentDays Churn.Yes  0.10
4    MadeCallToRetentionTeam.No Churn.Yes -0.07
5   MadeCallToRetentionTeam.Yes Churn.Yes  0.07
6         TotalRecurringCharge Churn.Yes -0.06
7         HandsetWebCapable.No Churn.Yes  0.06
8        HandsetWebCapable.Yes Churn.Yes -0.06
9               RetentionCalls Churn.Yes  0.06
10           CreditRating.5.Low Churn.Yes -0.06
11              MonthlyMinutes Churn.Yes -0.05
12               ReceivedCalls Churn.Yes -0.04
13               PeakCallsInOut Churn.Yes -0.04
14            OffPeakCallsInOut Churn.Yes -0.04
15               HandsetModels Churn.Yes -0.04
16             UnansweredCalls Churn.Yes -0.03
17            CustomerCareCalls Churn.Yes -0.03
18               OutboundCalls Churn.Yes -0.03
19                InboundCalls Churn.Yes -0.03
20            CallWaitingCalls Churn.Yes -0.03
21                    Handsets Churn.Yes -0.03
22                      AgeHH1 Churn.Yes -0.03
23         HandsetRefurbished.No Churn.Yes -0.03
```



```
24      HandsetRefurbished.Yes Churn.Yes  0.03
25    RetentionOffersAccepted Churn.Yes  0.03
26               HandsetPrice Churn.Yes -0.03
27             OverageMinutes Churn.Yes  0.02
28               ThreewayCalls Churn.Yes -0.02
29             MonthsInService Churn.Yes  0.02
30                      AgeHH2 Churn.Yes -0.02
31          BuysViaMailOrder.No Churn.Yes  0.02
32         BuysViaMailOrder.Yes Churn.Yes -0.02
33      RespondsToMailOffers.No Churn.Yes  0.02
34     RespondsToMailOffers.Yes Churn.Yes -0.02
35       CreditRating.1.Highest Churn.Yes  0.02
36          CreditRating.2.High Churn.Yes  0.02
37          CreditRating.3.Good Churn.Yes  0.02
38        CreditRating.4.Medium Churn.Yes -0.02
39              MonthlyRevenue Churn.Yes -0.01
40                RoamingCalls Churn.Yes  0.01
41                DroppedCalls Churn.Yes -0.01
42                BlockedCalls Churn.Yes -0.01
43          DroppedBlockedCalls Churn.Yes -0.01
```

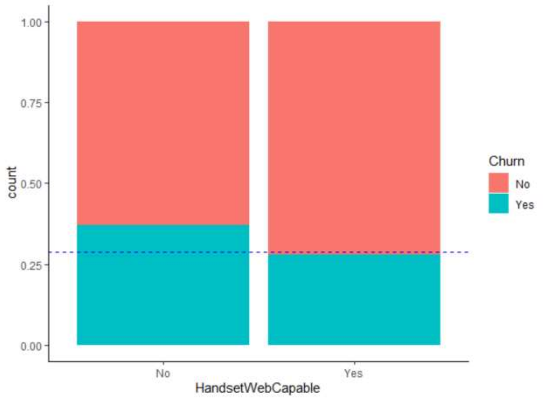


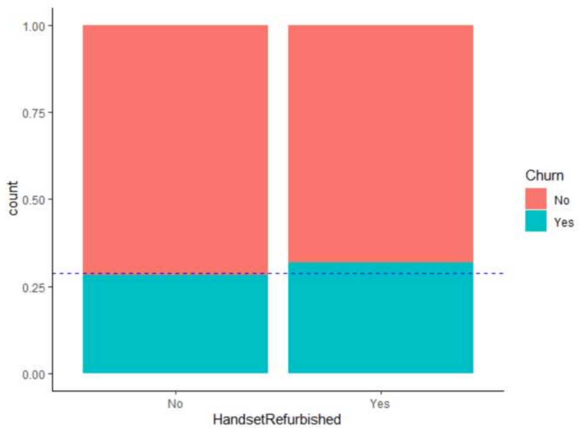
```
44                  ActiveSubs Churn.Yes  0.01
45              ChildrenInHH.No Churn.Yes -0.01
46             ChildrenInHH.Yes Churn.Yes  0.01
47                TruckOwner.No Churn.Yes  0.01
48               TruckOwner.Yes Churn.Yes -0.01
49              HasCreditCard.No Churn.Yes  0.01
50             HasCreditCard.Yes Churn.Yes -0.01
51            NewCellphoneUser.No Churn.Yes  0.01
52           NewCellphoneUser.Yes Churn.Yes -0.01
53      ReferralsMadeBySubscriber Churn.Yes -0.01
54                  IncomeGroup Churn.Yes -0.01
55              OwnsMotorcycle.No Churn.Yes -0.01
56             OwnsMotorcycle.Yes Churn.Yes  0.01
57          CreditRating.6.VeryLow Churn.Yes -0.01
58              PrizmCode.Other Churn.Yes -0.01
59              PrizmCode.Rural Churn.Yes  0.01
60           PrizmCode.Suburban Churn.Yes -0.01
61               PrizmCode.Town Churn.Yes  0.01
62            Occupation.Crafts Churn.Yes -0.01
63         Occupation.Homemaker Churn.Yes  0.01
```

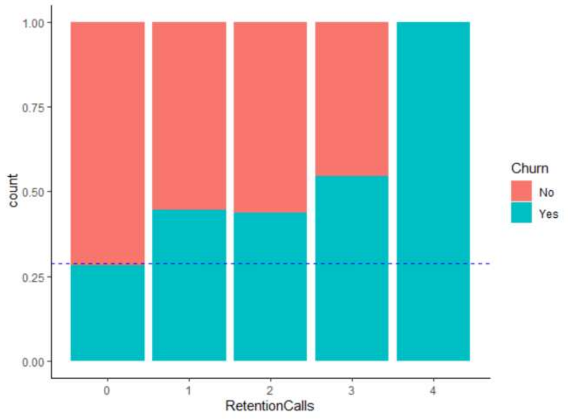


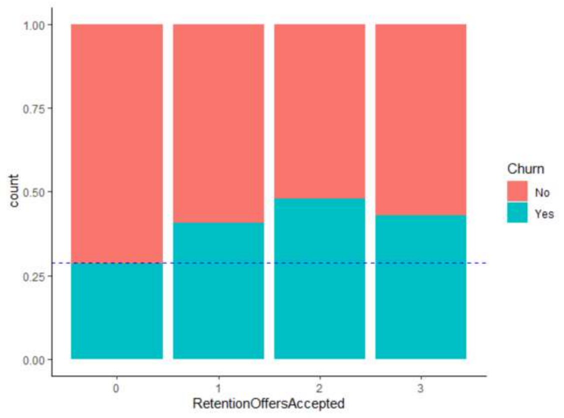
```
64             Occupation.Other Churn.Yes  0.01
65      Occupation.Professional Churn.Yes -0.01
66           Occupation.Retired Churn.Yes -0.01
67             MaritalStatus.No Churn.Yes  0.01
68            MaritalStatus.Yes Churn.Yes -0.01
69          CallForwardingCalls Churn.Yes  0.00
70                   RVOwner.No Churn.Yes  0.00
71                  RVOwner.Yes Churn.Yes  0.00
72             OptOutMailings.No Churn.Yes  0.00
73            OptOutMailings.Yes Churn.Yes  0.00
74               OwnsComputer.No Churn.Yes  0.00
75              OwnsComputer.Yes Churn.Yes  0.00
76         CreditRating.7.Lowest Churn.Yes  0.00
77          Occupation.Clerical Churn.Yes  0.00
78              Occupation.Self Churn.Yes  0.00
79           Occupation.Student Churn.Yes  0.00
```

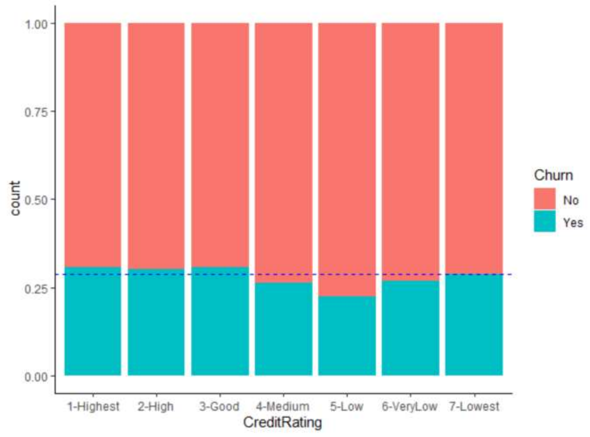
1.3 - Other Top Correlation Graphs:

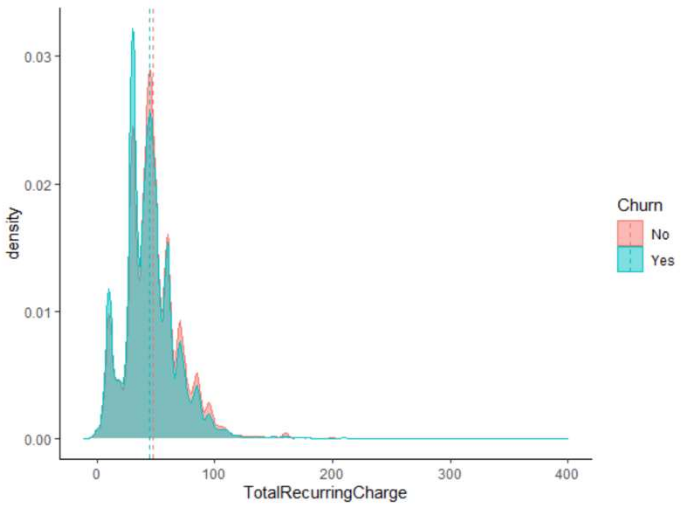


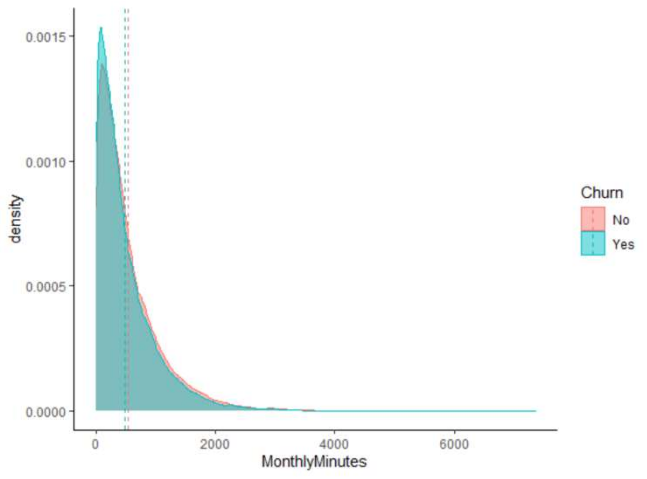




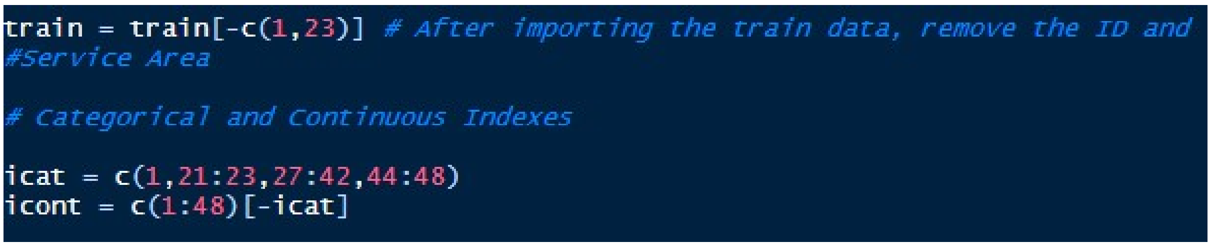




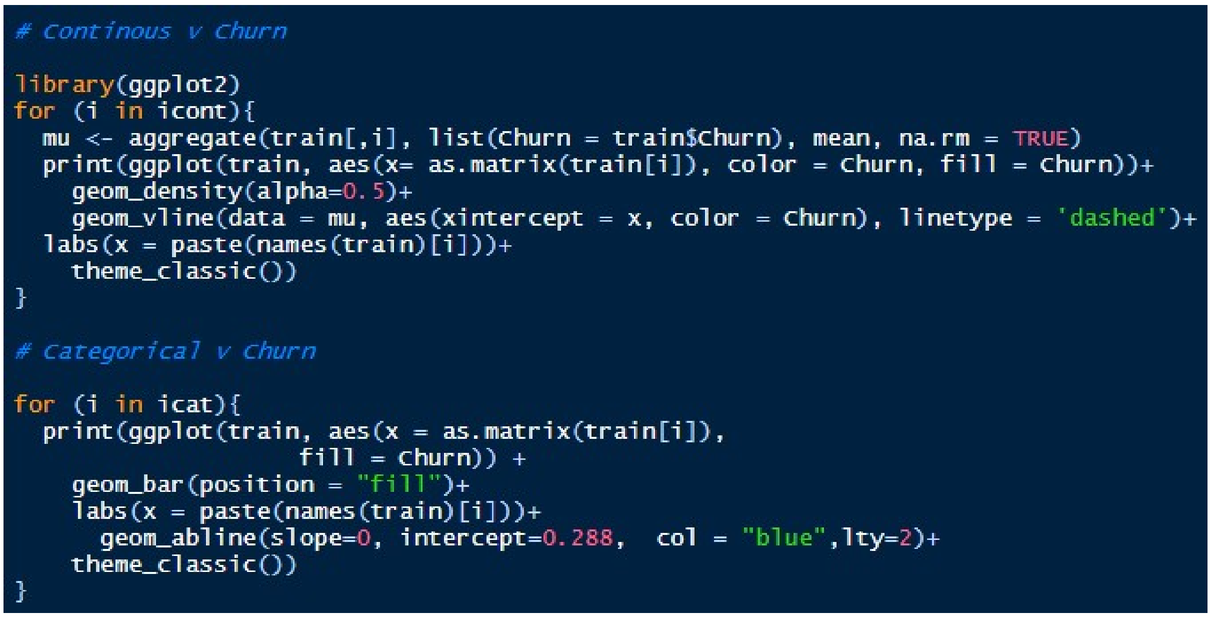




Code Set-up For Plotting All Densities:



```r
train = train[-c(1,23)] # After importing the train data, remove the ID and
#Service Area

# Categorical and Continuous Indexes

icat = c(1,21:23,27:42,44:48)
icont = c(1:48)[-icat]
```

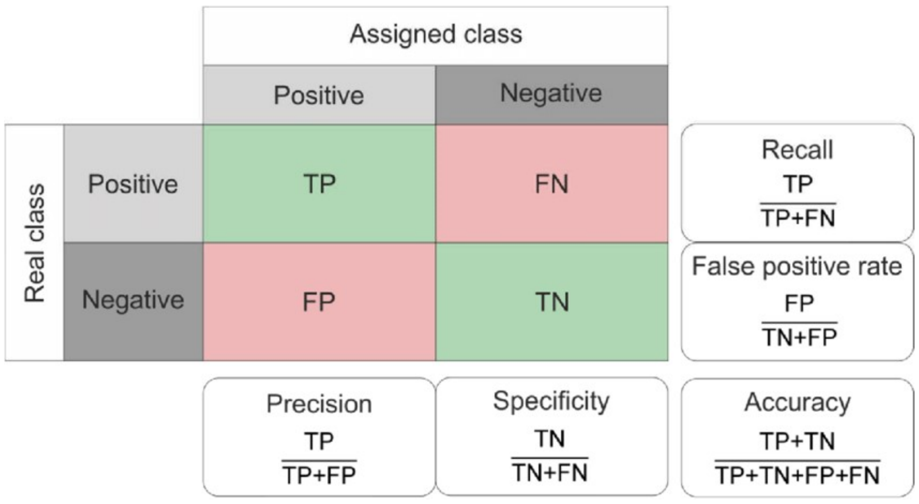


```r
# Continous v Churn

library(ggplot2)
for (i in icont){
  mu <- aggregate(train[,i], list(Churn = train$Churn), mean, na.rm = TRUE)
  print(ggplot(train, aes(x= as.matrix(train[i]), color = Churn, fill = Churn))+
    geom_density(alpha=0.5)+
    geom_vline(data = mu, aes(xintercept = x, color = Churn), linetype = 'dashed')+
  labs(x = paste(names(train)[i]))+
    theme_classic())
}

# Categorical v Churn

for (i in icat){
  print(ggplot(train, aes(x = as.matrix(train[i]),
                    fill = Churn)) +
    geom_bar(position = "fill")+
    labs(x = paste(names(train)[i]))+
      geom_abline(slope=0, intercept=0.288,  col = "blue",lty=2)+
    theme_classic())
}
```

2 – Choosing the Optimal Model

2.1 – Metrics Used

The following diagram illustrates the relative measures based on the predicted and actual classes. Note that recall = sensitivity.


Assigned class
Positive
Negative
Real class
Positive
Negative
TP
FN
FP
TN
Recall $\frac{TP}{TP+FN}$
False positive rate $\frac{FP}{TN+FP}$
Precision $\frac{TP}{TP+FP}$
Specificity $\frac{TN}{TN+FN}$
Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

Source: https://www.researchgate.net/figure/Confusion-matrix-and-related-performance-measures_fig2_328860691

<u>Precision:</u>

Precision is a measure of the proportion of true positives predicted out of the total amount predicted for the positive class. This is of utmost importance to this analysis as the goal is to predict 3000 'Yes' (positive) class and have the highest % of these correct as possible.

<u>Recall/Sensitivity:</u>

Recall is a measure of the proportion of positive cases that are identified. This is also a useful metric in this case as we can measure how many churn customers were identified when predicting 3000 as 'Yes'.

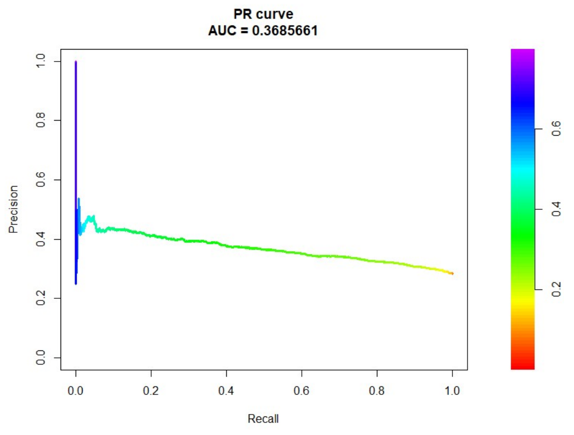
<u>Specificity:</u>

Specificity is a measure of the proportion of negative classes that are identified. This is of less importance to this investigation as we are focusing on the 'Yes' class, however this metric still explains the general performance of the classifier and therefore has similar recommendations as the other metrics, albeit not exactly the same.

<u>Precision-Recall Curve:</u>

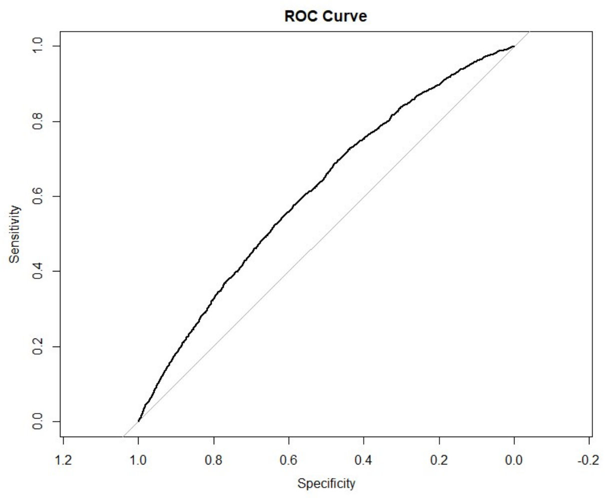
The precision-recall curve plots precision and recall for each threshold probability. Therefore, as these are two positive class specific measures, the area under the curve gives good indication of how the classifier performs relative to the positive class. As the 'Yes' class is paramount in this investigation, this area will maximise the classifier's ability to classify the positive class 'Yes'.

The baseline (if classified at random) to compare the AUC_PR is the proportion of ‘Yes’ cases in the data. This is approximately 0.288 in this scenario. AUC_PR>0.288 is an improvement over random classification and the greater the better. An example plot is shown to the right, where the gradient bar refers to the probability threshold used.



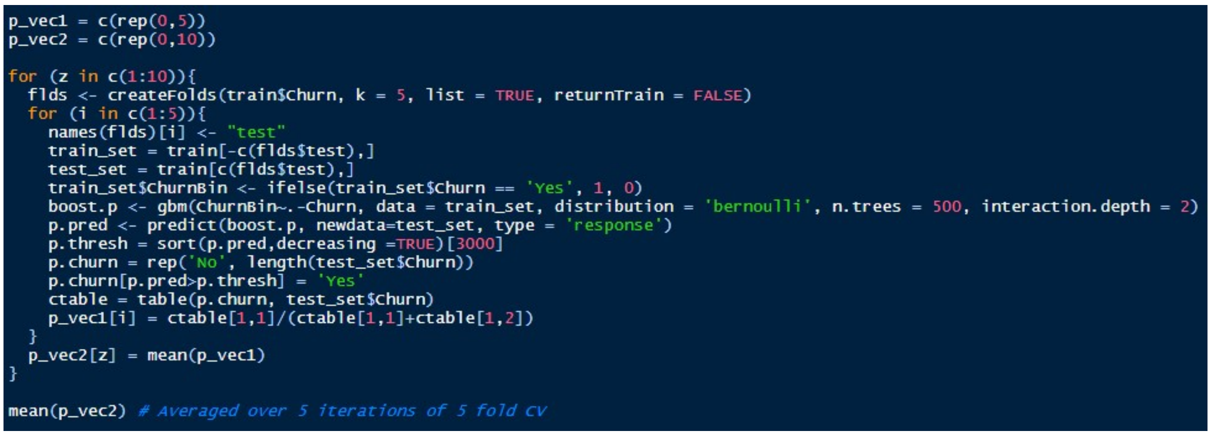
ROC Curve:

The ROC curve plots the specificity and recall for each threshold probability. This gives a great measure for the general performance of the classifier, but tuning based on the AUC_ROC may sacrifice precision for specificity, which is less important in this investigation. Therefore while it will not be used to tune the model, it is still a useful measure to report.



The baseline (if classified at random) to compare the AUC_ROC is 0.5 irrespective of the scenario. AUC_ROC>0.5 is an improvement over random classification and the greater the better. An example plot is to the right, where the grey line is the baseline.

2.2 – Precision Cross Validation

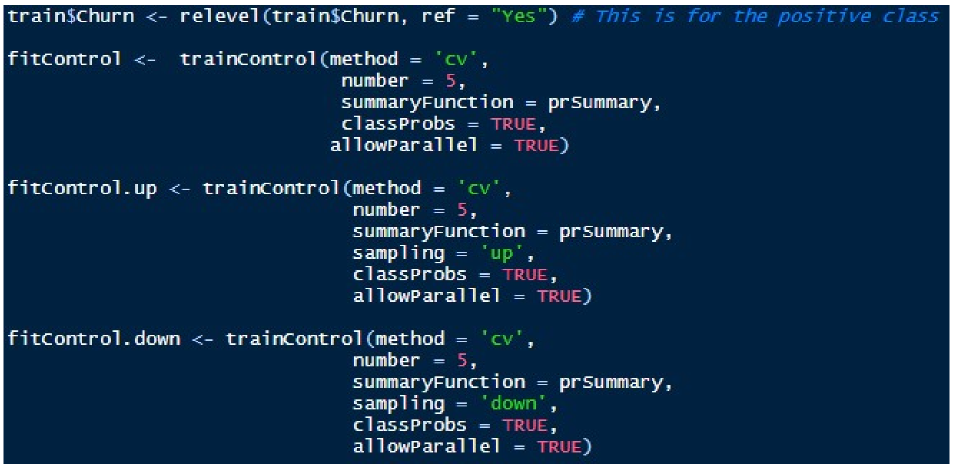


```
p_vec1 = c(rep(0,5))
p_vec2 = c(rep(0,10))

for (z in c(1:10)){
  flds <- createFolds(train$Churn, k = 5, list = TRUE, returnTrain = FALSE)
  for (i in c(1:5)){
    names(flds)[i] <- "test"
    train_set = train[-c(flds$test),]
    test_set = train[c(flds$test),]
    train_set$ChurnBin <- ifelse(train_set$Churn == 'Yes', 1, 0)
    boost.p <- gbm(ChurnBin~.-Churn, data = train_set, distribution = 'bernoulli', n.trees = 500, interaction.depth = 2)
    p.pred <- predict(boost.p, newdata=test_set, type = 'response')
    p.thresh = sort(p.pred,decreasing =TRUE)[3000]
    p.churn = rep('No', length(test_set$Churn))
    p.churn[p.pred>p.thresh] = 'Yes'
    ctable = table(p.churn, test_set$Churn)
    p_vec1[i] = ctable[1,1]/(ctable[1,1]+ctable[1,2])
  }
  p_vec2[z] = mean(p_vec1)
}

mean(p_vec2) # Averaged over 5 iterations of 5 fold CV
```

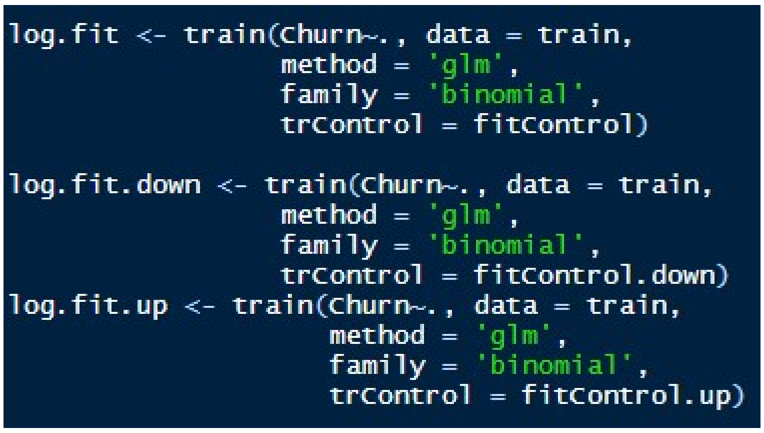
2.3 – Model Cross Validation and Grid Search

Cross-Validation:

Cross-Validation is a versatile approach to predicting the out-of-sample performance of a classifier. For this analysis we utilise cross-validation with 5-folds, therefore the data is shuffled and split into 5 folds randomly. In each iteration the model is trained on 4/5 folds and test scores, such as AUC, is calculated on the remaining fold. This is repeated 5 times, once for each fold, and the final score is the averaged result. In R this can be easily achieved through the ‘caret’ package. For this analysis cross-validation was done for each model type with each type of sampling. Example code when no grid-search is required:



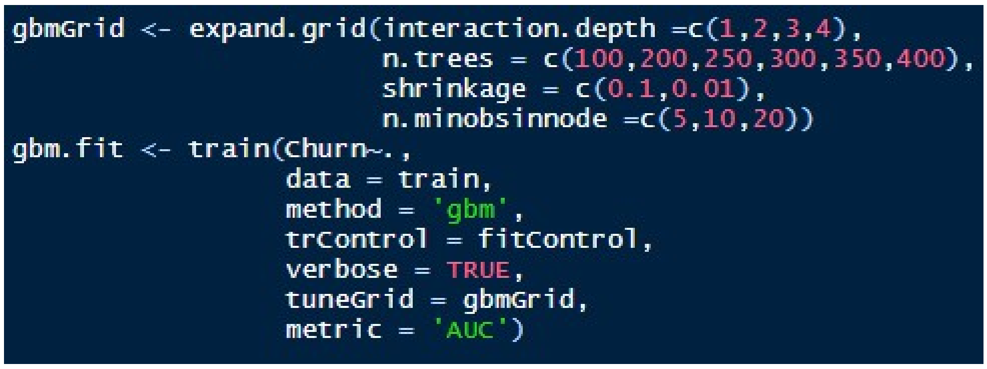
```r
train$Churn <- relevel(train$Churn, ref = "Yes") # This is for the positive class

fitControl <-  trainControl(method = 'cv',
                            number = 5,
                            summaryFunction = prSummary,
                            classProbs = TRUE,
                           allowParallel = TRUE)

fitControl.up <- trainControl(method = 'cv',
                              number = 5,
                              summaryFunction = prSummary,
                              sampling = 'up',
                              classProbs = TRUE,
                              allowParallel = TRUE)

fitControl.down <- trainControl(method = 'cv',
                              number = 5,
                              summaryFunction = prSummary,
                              sampling = 'down',
                              classProbs = TRUE,
                              allowParallel = TRUE)
```



```r
log.fit <- train(Churn~., data = train,
                 method = 'glm',
                 family = 'binomial',
                 trControl = fitControl)

log.fit.down <- train(Churn~., data = train,
                 method = 'glm',
                 family = 'binomial',
                 trControl = fitControl.down)
log.fit.up <- train(Churn~., data = train,
                    method = 'glm',
                    family = 'binomial',
                    trControl = fitControl.up)
```

<u>Grid Search:</u>

Grid-search utilises cross-validation, training a model 5 times (for 5 folds) for each combination of hyperparameters set. To do this in R the ‘caret’ package is utilised. We must first set up a grid and then add this to the train() function in caret, similar to the regular cross-validation. Note that this is computationally intensive for a high amount of combinations, so only theoretically feasible combinations are tested. Example code for boosted trees:



```r
gbmGrid <- expand.grid(interaction.depth =c(1,2,3,4),
                       n.trees = c(100,200,250,300,350,400),
                       shrinkage = c(0.1,0.01),
                       n.minobsinnode =c(5,10,20))
gbm.fit <- train(Churn~.,
                 data = train,
                 method = 'gbm',
                 trControl = fitControl,
                 verbose = TRUE,
                 tuneGrid = gbmGrid,
                 metric = 'AUC')
```

2.4 - Relative Method Results

Logistic Regression:

- The logistic regression is a generalised linear model often used for modelling a binary response.
- The x values are mapped between 0,1 such that the probability of an event occurring can be modelled and predicted.
- As the logistic regression is a generalised linear model it has a linear decision boundary and therefore is relatively inflexible. This is true for all the proceeding extensions as well.
- The logistic regression formula is as follows, with the coefficients being estimated through maximum likelihood estimators:

$$\underbrace{\log\left(\frac{P(x_i)}{1 - P(x_i)}\right)}_{\text{Log odds}} = \mathbf{x}_i^\top \beta$$

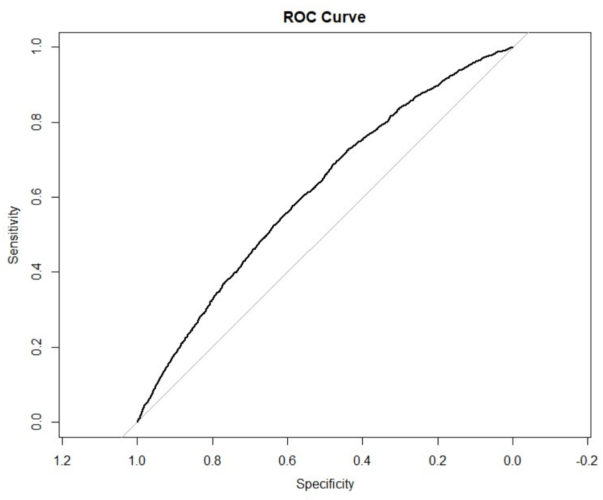
- Modelling Results:

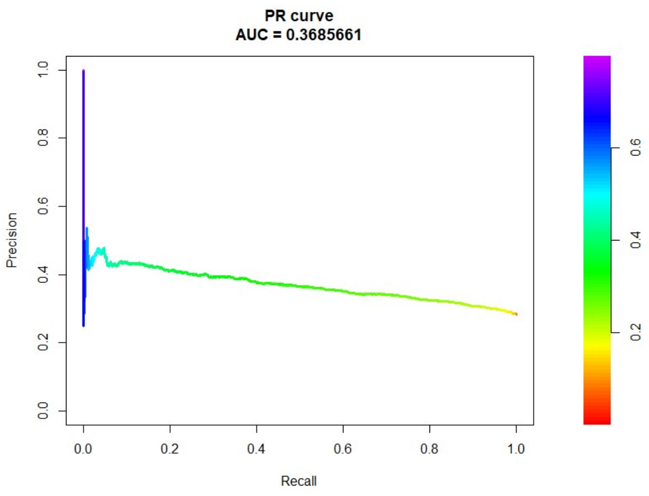
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.3703363	0.6075408	0.3783	0.3991	None

- Confusion Matrix:

	Reference		
Prediction	Yes	No	Total
Yes	1135	1865	3000
No	1709	5291	7000
Total	2844	7156	

- ROC and PR Curves:





Subset Logistic Regression:

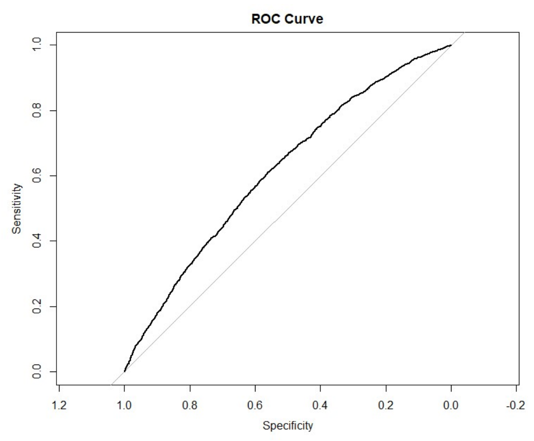
- Subset selection identifies a subset of p predictors that are believed to be related to the response.
- For this application, a hybrid approach was used where predictors are added/deleted in steps until the 'best' model is found based on a certain criterion. The criteria used in this case was AIC.
- This is applied to the logistic regression as the response being modelled is binary.
- Modelling Results:

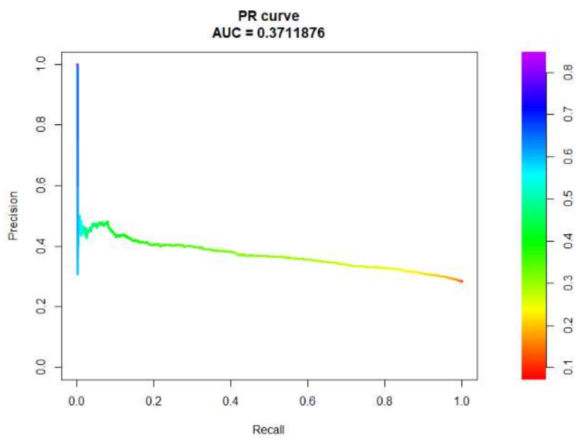
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.3702763	0.6070441	0.3798	0.4005	None

- Confusion Matrix:

	Reference		
Prediction	Yes	No	Total
Yes	1140	1860	3000
No	1704	5296	7000
Total	2844	7156	

- ROC and PR Curves:





- The predictors specified by the stepwise algorithm to be most statistically important are: MonthlyRevenue, MonthlyMinutes, TotalRecurringCharge, DroppedCalls, MonthsInService,Handsets,CurrentEquipmentDays,AgeHH1,ChildrenInHH,HandsetRefurbished,HandsetWebCapable,RespondsToMailOffers,MadeCallToRetentionTeam,CreditRating,PrizmCode.
- The results for the subset selection are very similar to the original logistic regression with all predictors.

Logistic Ridge Regression:

- Ridge regression is a shrinkage method that involves fitting all predictors but shrinking the estimates of coefficients towards 0.
- The magnitude of shrinkage is dependent on a tuning parameter lambda, which needs to be turned to find the optimal fit.
- Typically, the smaller the impact of the variable on the response, the smaller the resulting coefficient will be. However, in Ridge Regression parameters don't reach 0, only approach it.
- For a set lambda, the coefficients are found by minimising the following criterion:

$$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

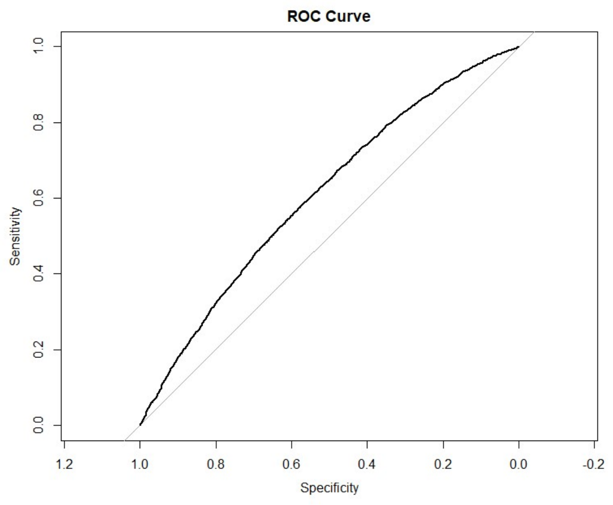
- Lambda = 0.008667139 as chosen by grid search.
- Modelling Results:

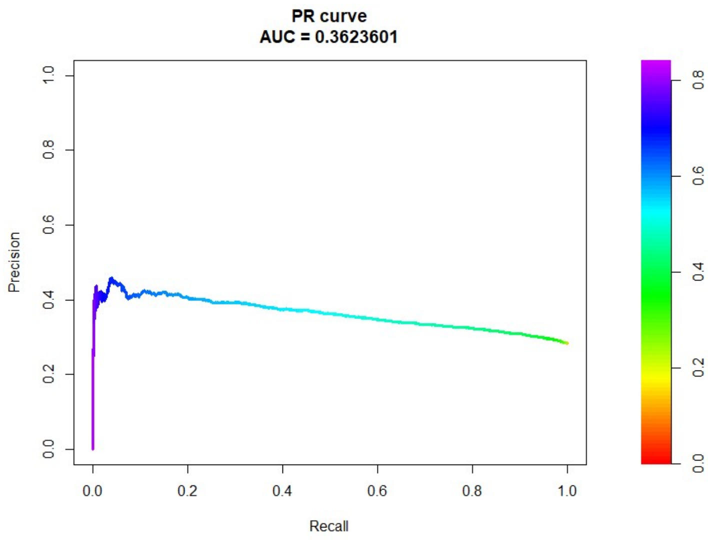
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.3697555	0.6075718	0.3745	0.3949	None

- Confusion Matrix:

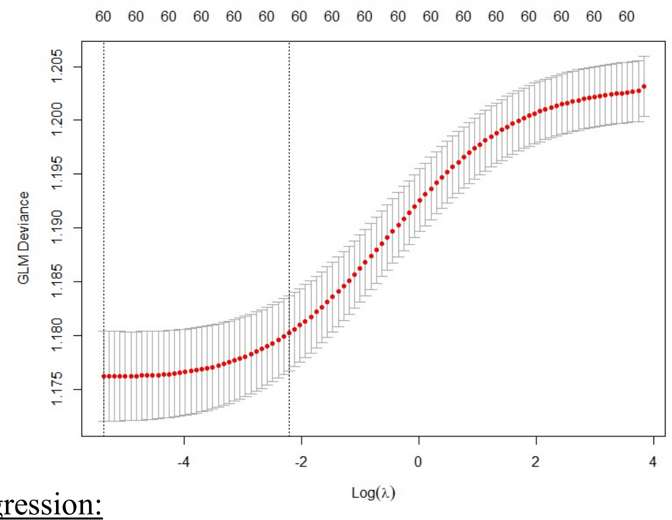
	Reference		
Prediction	Yes	No	Total
Yes	1124	1876	3000
No	1720	5280	7000
Total	2844	7156	

- ROC and PR Curves:





- The following a plot of log(lambda) against deviance, a measure of goodness of fit. Here it it obvious a low penalty results in the best model in this scenario.



Logistic Lasso Regression:

- Lasso regression is a shrinkage method that involves fitting all predictors but shrinking the estimates of coefficients towards (and possibly equalling) 0.
- The magnitude of shrinkage is dependent on a tuning parameter lambda, which needs to be turned to find the optimal fit.
- Typically, the smaller the impact of the variable on the response, the smaller the resulting coefficient will be. However, in Lasso Regression parameters can equal 0, resulting in them being excluded from the model. Therefore Lasso Regression can be used for variable selection.
- For a set lambda, the coefficients are found by minimising the following criterion:

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

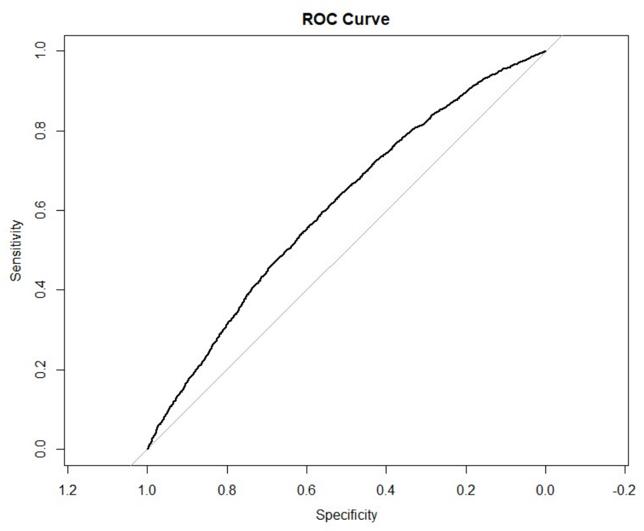
- Lambda = 0.0002785414
- Modelling Results:

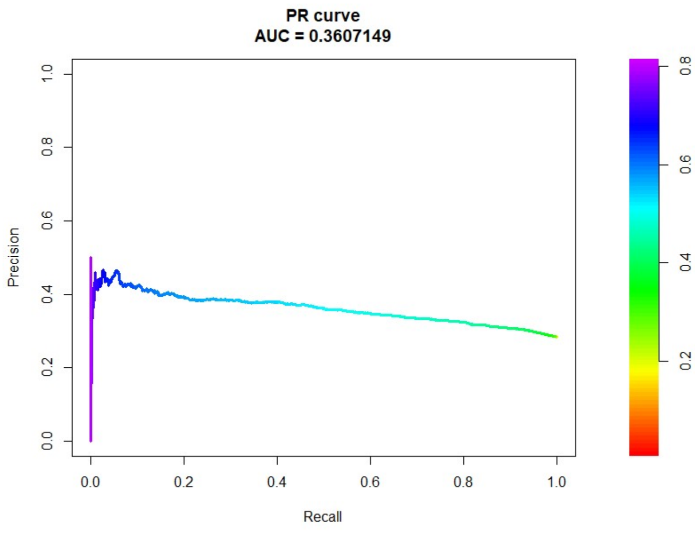
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.37034585	0.6072644	0.3788	0.3994	None

- Confusion Matrix:

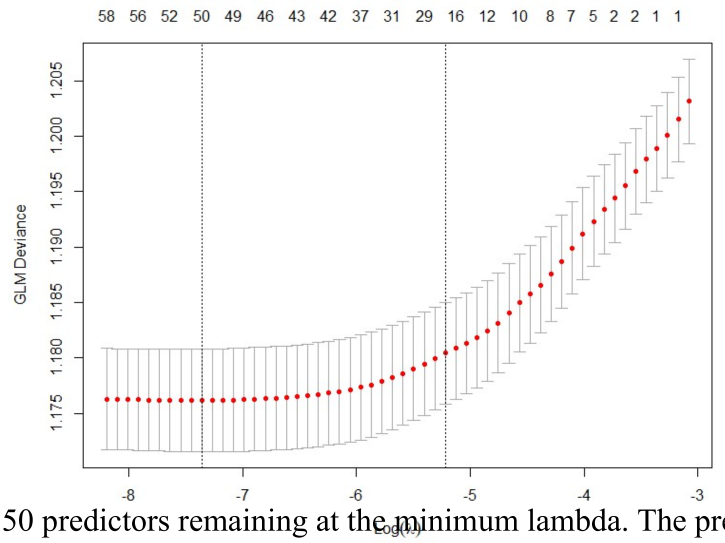
	Reference		
Prediction	Yes	No	Total
Yes	1137	1863	3000
No	1707	5293	7000
Total	2844	7156	

ROC and PR Curves:





- Here it is obvious an even smaller penalty is required to attain the minimum deviance. It is often good practice to utilise the lambda that is 1 standard deviation away from the minimum as a form of tradeoff to increase the interpretability due to less predictors.



- There are 50 predictors remaining at the minimum lambda. The predictors **removed** at the minimum lambda are: BlockedCalls, OutboundCalls, OffPeakCallsInOut, CallForwardingCalls, TrunkOwner, BuysViaMailOrder, NewCellphoneUser, IncomeGroup, OccupationRetired, OccupationOther
- There are 23 predictors remaining at the 1sd lambda. The predictors **retained** at the 1sd lambda are: MonthlyMinutes, TotalRecurringCharge, OverageMinutes, DroppedCalls, CustomerCareCalls, ThreeWayCalls, DroppedBlockedCalls, MonthsInService, CurrentEquipmentDays, AgeHH1,AgeHH2, ChildreninHH, HandsetRefurbished, HandsetWebCapable, RespondsToMailOffers, RetentionCalls, HandsetPrice, MadeCallToRetentionTeam, CreditRating, PrizmCode.

Generalised Additive Models (GAM):

- GAM are an extension of the generalised linear model, allowing for the covariates to be modelled by non-linear functions (often splines) rather than singular coefficients.
- The additive structure of linear models is maintained, however this results in the assumption of no interaction between terms, as with the general linear model.
- A GAM follows the formula:

$$g(\mu_i) = \sum_{j=1}^{k} f_j(x_{ij})$$

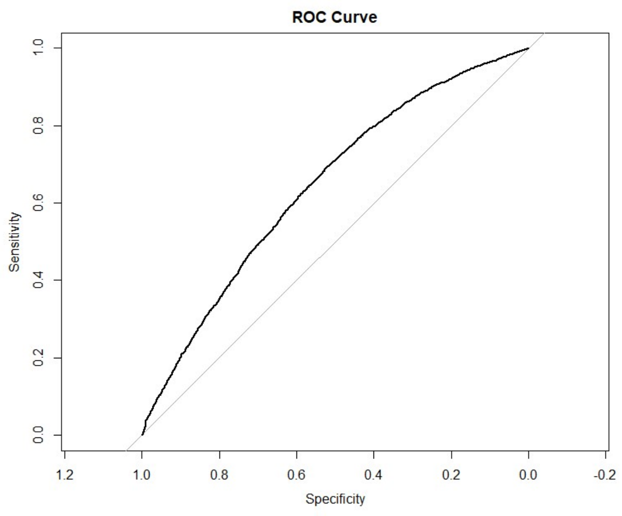
- Modelling Results:

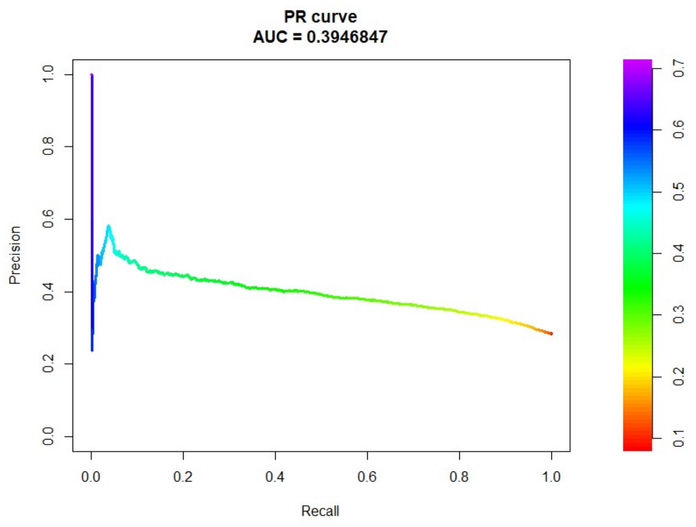
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.3847211	0.6275906	0.4023	0.4244	No

- Confusion Matrix:

	Reference		
Prediction	Yes	No	Total
Yes	1207	1793	3000
No	1637	5363	7000
Total	2844	7156	

- ROC and PR Curves:





- The GAM model seems to have a substantial performance increase over all the other generalised linear model extensions. This is presumably due to the ability for GAM to incorporate non-linear trends which are common in the data (as will be shown later in the report). However the linear/additive structure still seems to be of detriment.

<u>K Nearest Neighbours (KNN):</u>

- KNN is a simple non-parametric approach that considers the k nearest datapoints in the training set and assigns the new data point based on a majority vote.
- Due to the non-parametric nature the decision boundary is non-linear and no assumption is made on the distribution of the response.
- KNN typically uses Euclidean distance as a closeness measure, and therefore tends to struggle from the curse of dimensionality. This results in depleting performance as amount of predictors increase.
- The amount of neighbours, K, is a hyperparameter that needs to be tuned and determines how many datapoints are considered.
- The formula for probabilities from KNN is:

$$f_k(x) = \frac{1}{K} \sum_{x_i \in N_0} \mathbb{I}(y_i = k)$$

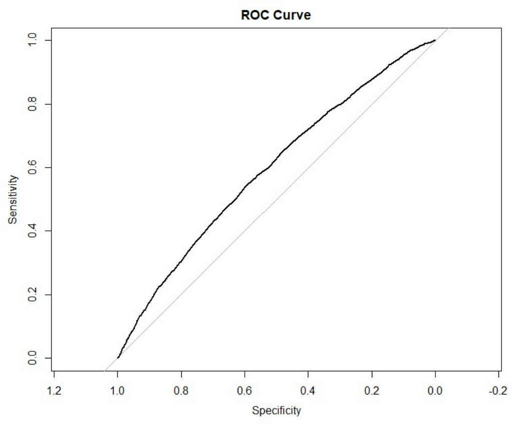
- K = 170 was chosen through grid search.
- Modelling Results:

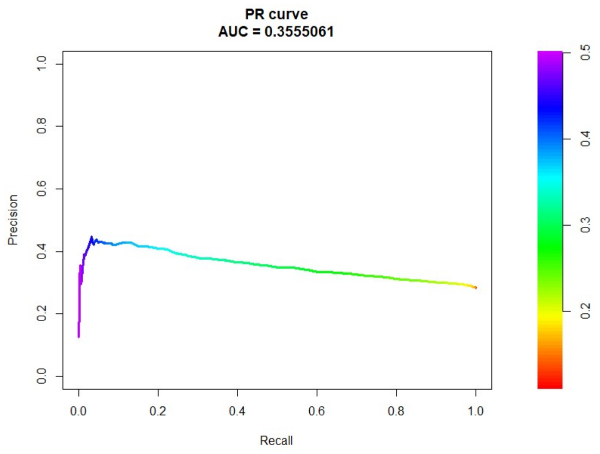
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.3683724	0.5990390	0.3647	0.3994	No

- Confusion Matrix (Note due to many predictions having the same probability 3000 exactly cannot be chosen without random choice):

	Reference		
Prediction	Yes	No	Total
Yes	1136	1979	3115
No	1708	5177	6885
Total	2844	7156	

- ROC and PR Curves:





- KNN is the second worst performing classifier tested despite the flexible nature. This is presumably due to the curse of dimensionality as mentioned before.

Decision Tree:

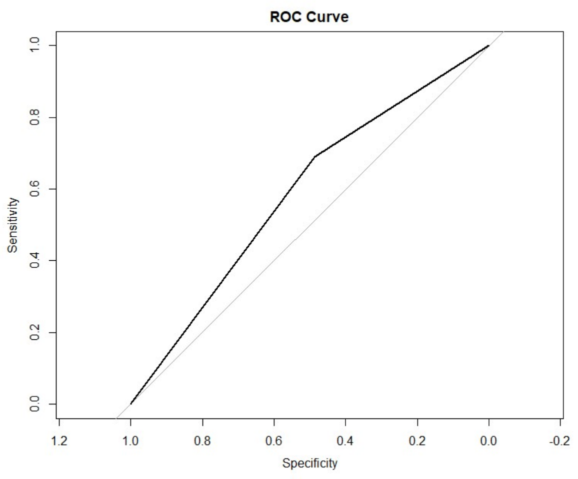
- Decision trees partition the feature space in way that minimises variance within the different split areas.
- When a datapoint is introduced, it goes down the decision tree paths until it ends at the end of one of the branches.
- At this point the datapoint is assigned to the majority class that from the training set that is in that partition.
- Decision trees by themselves are typically weak at prediction, but this can be improved through their extensions which will also be explored.
- Modelling Results:

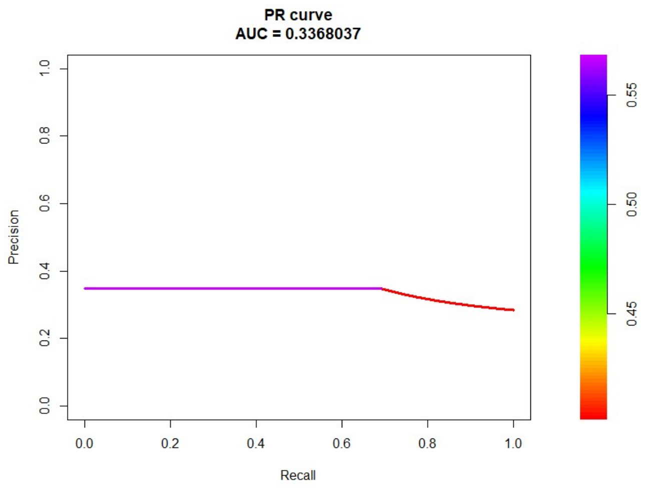
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.3368037	0.5889	0.3486	0.6913	Up

- Confusion Matrix:

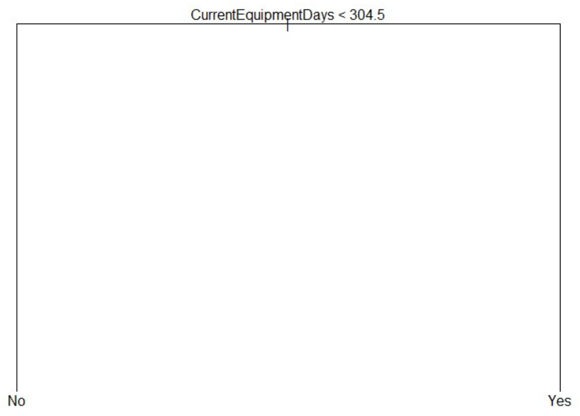
	Reference		
Prediction	Yes	No	Total
Yes	1966	3674	5640
No	878	3482	4360
Total	2844	7156	

- ROC and PR Curves:





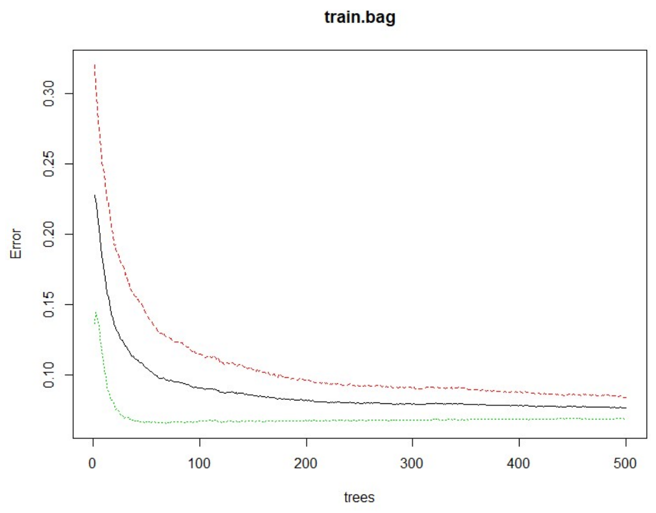
- Decision Trees have a nice graphical representation. Here is the graph for the tree grown for this scenario:



- It is obvious from this diagram that the most important variable to the tree is the CurrentEquipmentDays. While this singular tree is easy to interpret, it has weak predictive power. The single split also explains the elbow shape of the ROC and PR curves.

Bagging:

- Bagging is an extension on the simple decision tree.
- Bagging utilises B bootstrapped training sets to construct B decision trees. These trees are then averaged. However this B is not a critical hyperparameter as you cannot fit the amount of trees. Therefore an amount of trees are chosen where the training error levels out in the following graph.



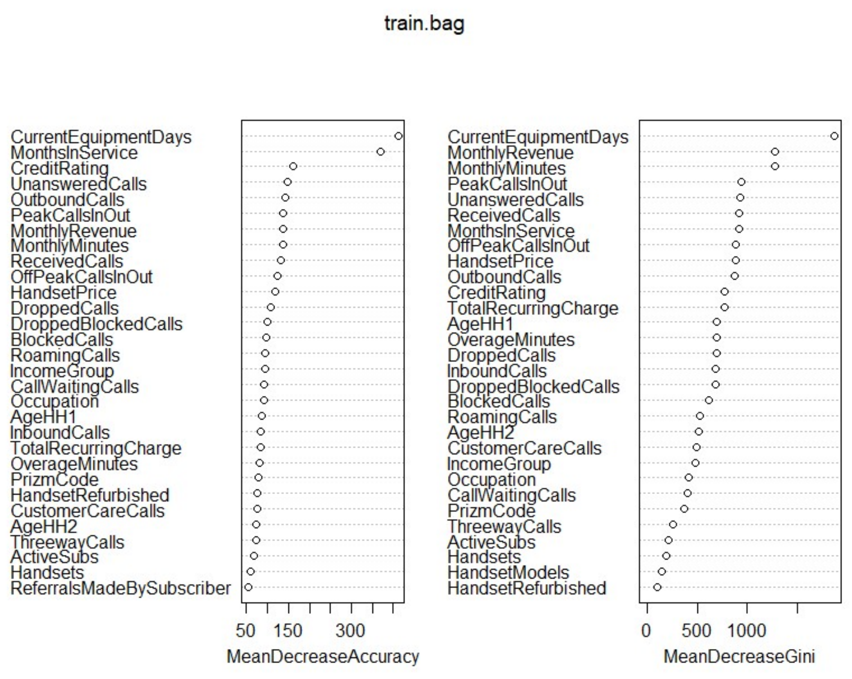
- In this case 300 trees is a suitable choice for B.
- For a given observation, we predict with each of the trees and take a majority vote on how many trees pick a certain class.
- Modelling Results:

PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.4198305	0.6468891	0.4074	0.4216	None

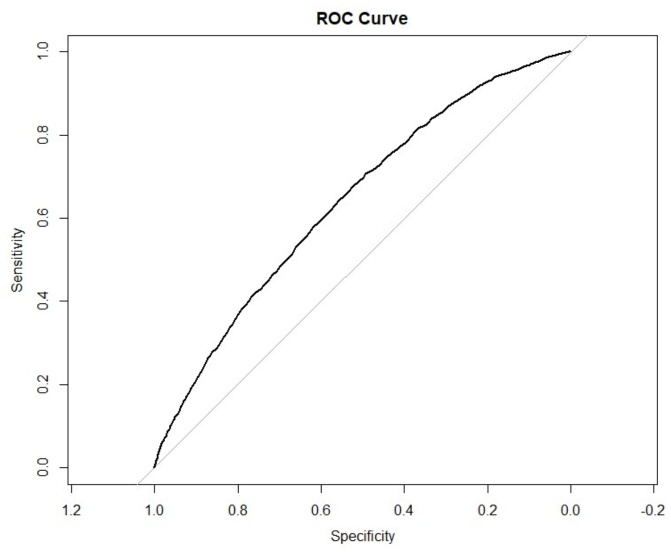
- Confusion Matrix (Note due to many predictions having the same probability 3000 exactly cannot be chosen without random choice):

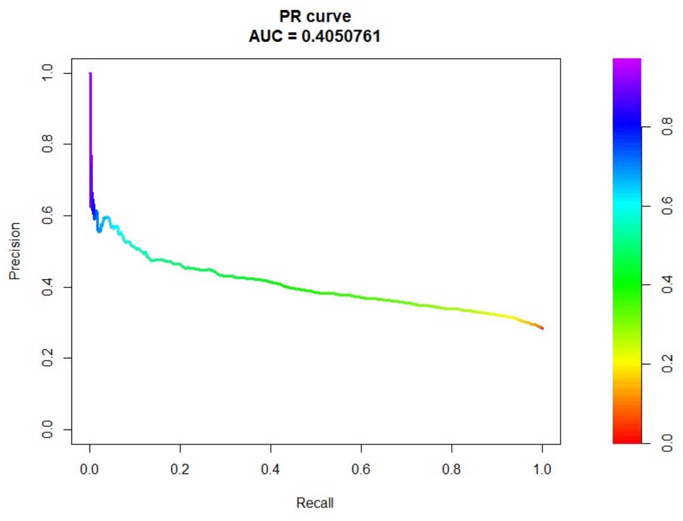
	Reference		
Prediction	Yes	No	Total
Yes	1199	1744	2943
No	1645	5412	7057
Total	2844	7156	

- It is obvious from the results that this is a drastic improvement in predictive power over a single tree. However, the interpretability is weakened as we cannot plot them simply like before. However for forests of trees we can calculate the relative importance of the variables through how much they contribute to decreasing variance (Gini) or decreasing accuracy. The relative importance for this scenario are plotted:



- ROC and PR Curves:





<u>Random Forest:</u>

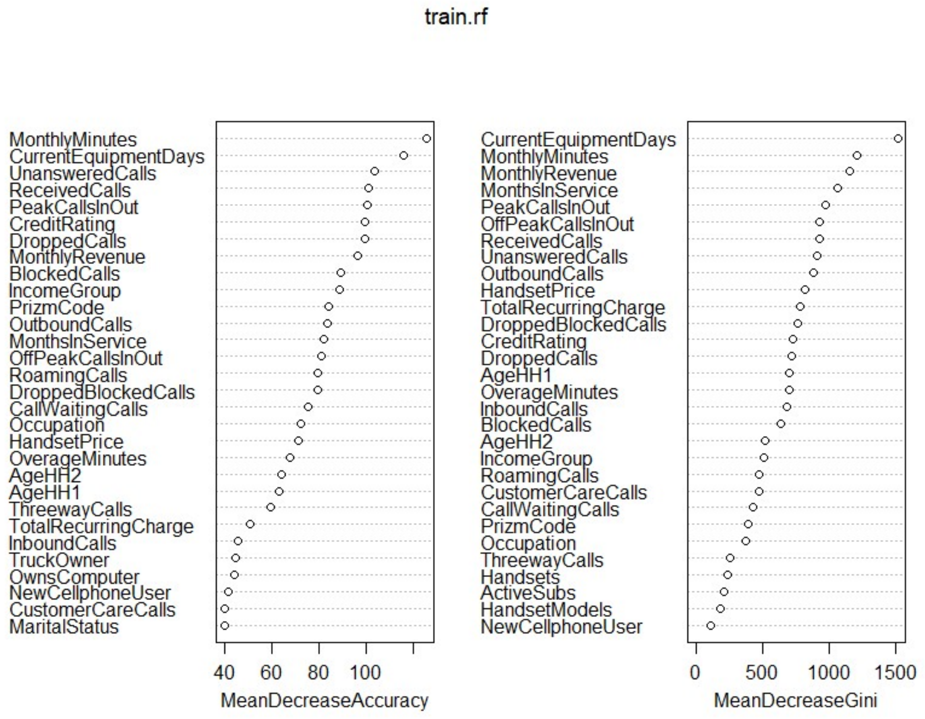
- Random forest is another extension on the simple decision tree.
- Random forests operate similarly to bagging, however there are only mtry amount of predictors allowed to be considered at each split. This creates trees that are highly differentiated, resulting in lower correlation between trees and therefore often better results.
- Hence, mtry needs to be tuned through gridsearch. In this case mtry = 10 was found to be optimal.
- The choice of 300 is also a suitable choice for B here due to the error leveling out at this B for random forest.
- Modelling Results:

PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.4251844	0.6546835	0.4253	0.4413	None

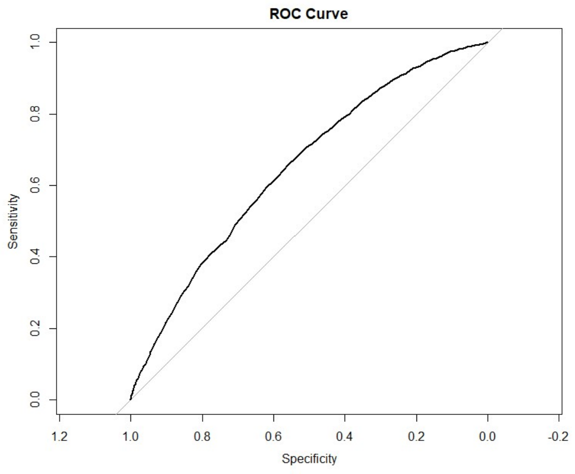
- Confusion Matrix (Note due to many predictions having the same probability 3000 exactly cannot be chosen without random choice):

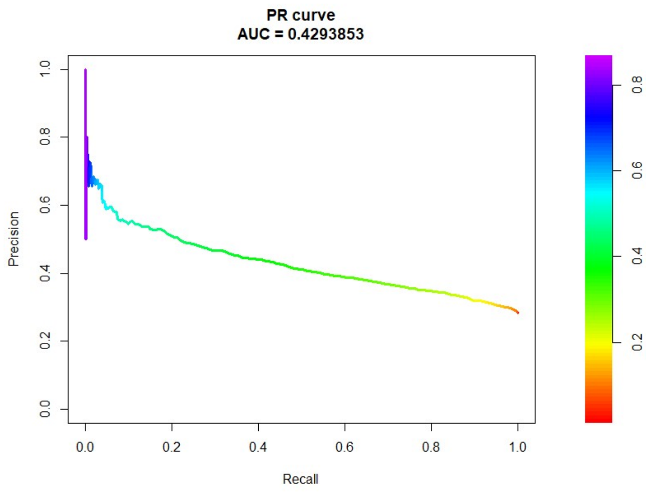
	Reference		
Prediction	Yes	No	Total
Yes	1236	1663	2951
No	1608	5493	7049
Total	2844	7156	

- The decrease in predictors allowed for each split resulted in better results in this case as seen through the results. Despite the lower interpretability relative to the decision tree, it is also possible to get the relative importance plots for random forest, as plotted:



- ROC and PR Curves:





Gradient Boosting Trees:

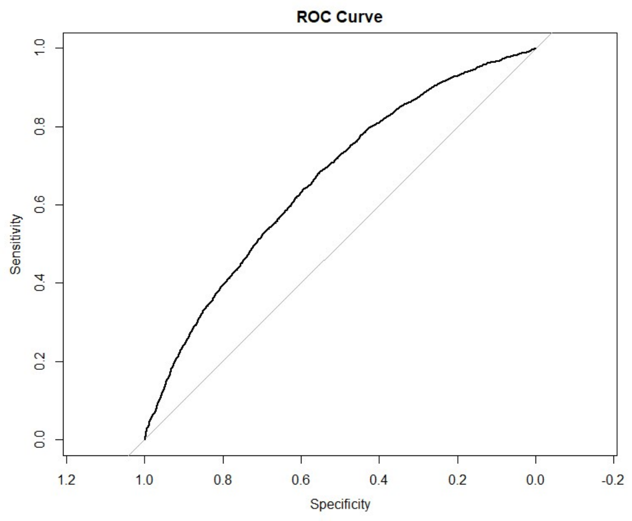
- Boosting is another extension to the simple decision tree, however it is different to bagging and random forests as trees are grown sequentially on the residuals of the previous model.
- This allows the model to slowly improve in the areas where it does not perform well.
- Due to this sequential nature, the amount of trees B needs to be tuned as there can be overfitting. In this case B=250 is the optimal amount of sequential trees.
- The depth (amount of decisions) of each tree needs to be tuned as well, with 4 being the optimal in this case.
- The shrinking parameter dictates how fast the model learns, which also needs to be tuned. Lambda = 0.01 is optimal in this case.
- Modelling Results:

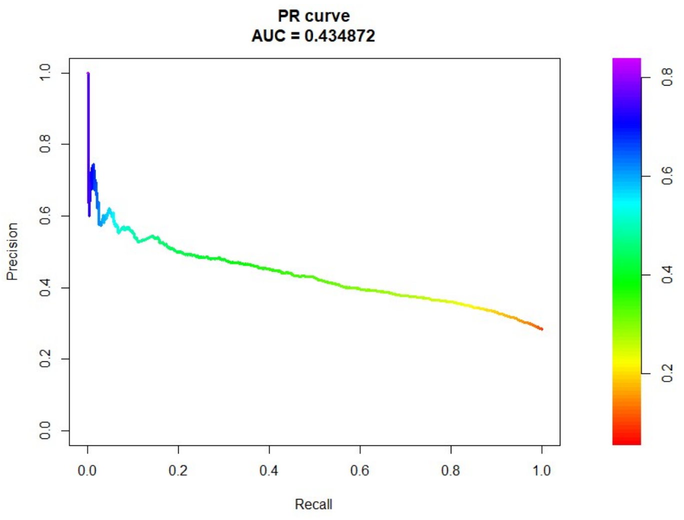
PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
0.4326199	0.6617765	0.4305	0.4539	None

- Confusion Matrix:

	Reference		
Prediction	Yes	No	Total
Yes	1292	1708	3000
No	1552	5448	7000
Total	2888	7112	

- ROC and RP Curves:





- Gradient Boosting trees are the best performing model out of all those considered. This indicates that the decision boundary is likely very non-linear. While it is not as easy to interpret as the decision tree, it is still possible to get relative predictor influences and to get marginal plots for the impact of each variable. These are discussed in the main report.

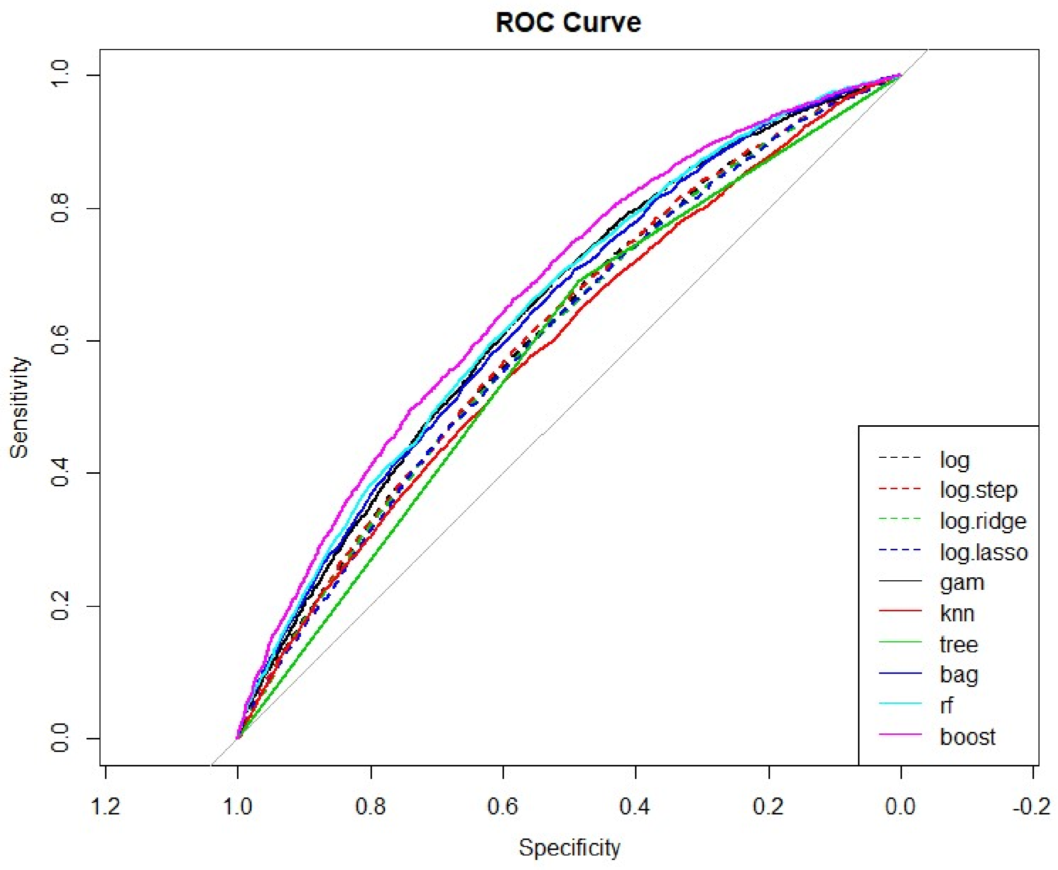
2.5– Summary of Model Performance:

Ordered (by PR_AUC) Metric Table:

Model	PR_AUC	ROC_AUC	Precision	Sensitivity	Sampling
Gradient Boost Trees	0.4326199	0.6617765	0.4305	0.4539	None
Random Forest	0.4251844	0.6546835	0.4253	0.4413	None
Bagging	0.4198305	0.6468891	0.4074	0.4216	None
GAM	0.3847211	0.6275906	0.4023	0.4244	None
Lasso	0.37034585	0.6072644	0.3788	0.3994	None
Logistic	0.3703363	0.6075408	0.3783	0.3991	None
Logistic Subset	0.3702763	0.6070441	0.3798	0.4005	None
Ridge	0.3697555	0.6075718	0.3745	0.3949	None
KNN	0.3683724	0.5990390	0.3647	0.3994	None
Decision Tree	0.3368037	0.5889	0.3486	0.6913	Up

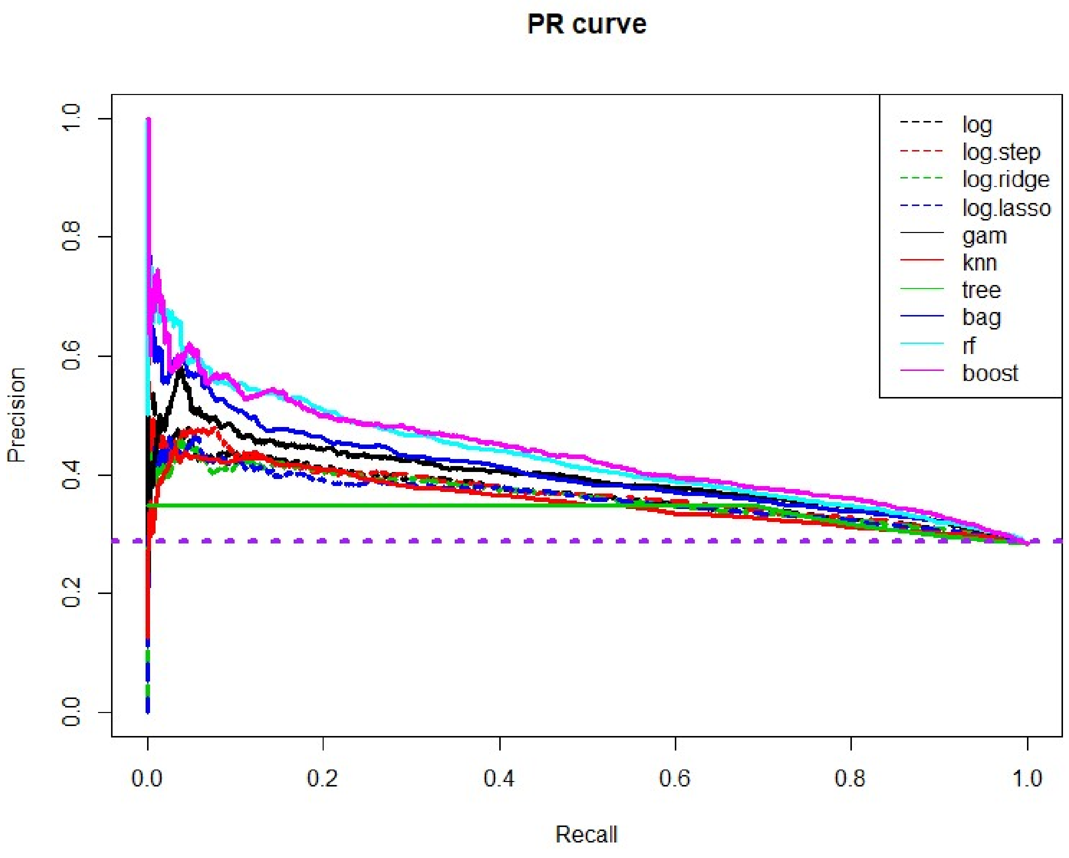
ROC Curves:

- Note the baseline is the grey 45* line, with AUC = 0.5.

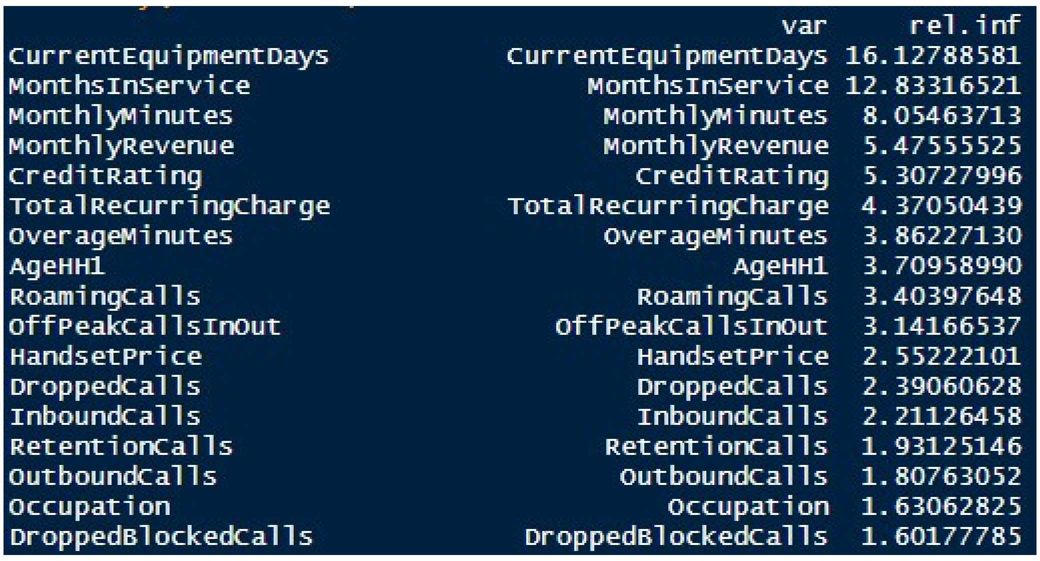


PR Curves:

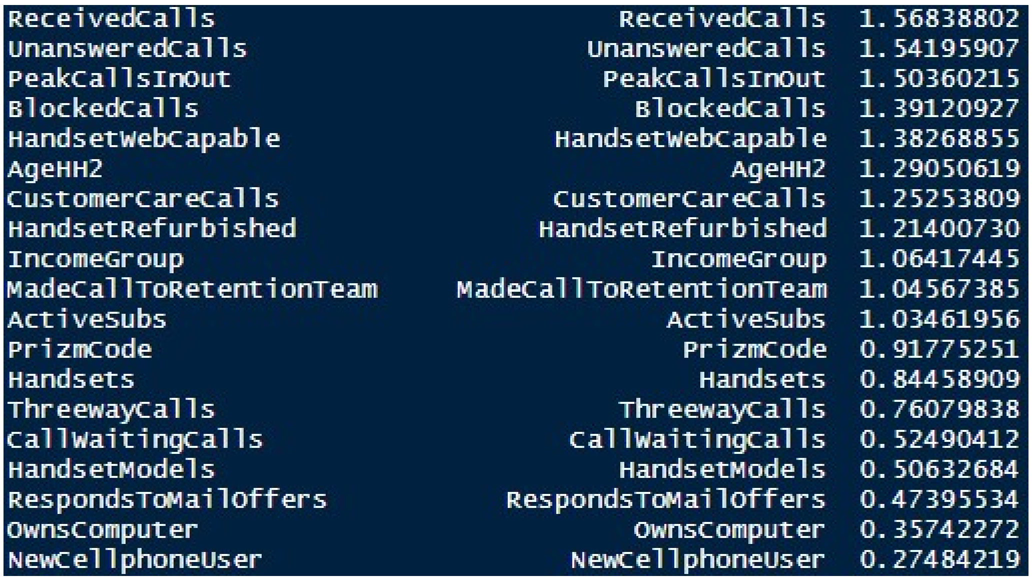
- Note the baseline is the dotted purple line at precision = 0.288 (proportion positive in dataset)



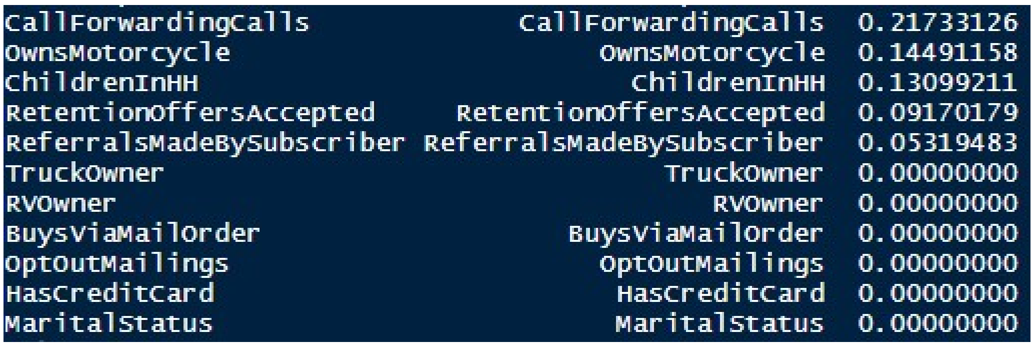
2.6 – Relative Influence List



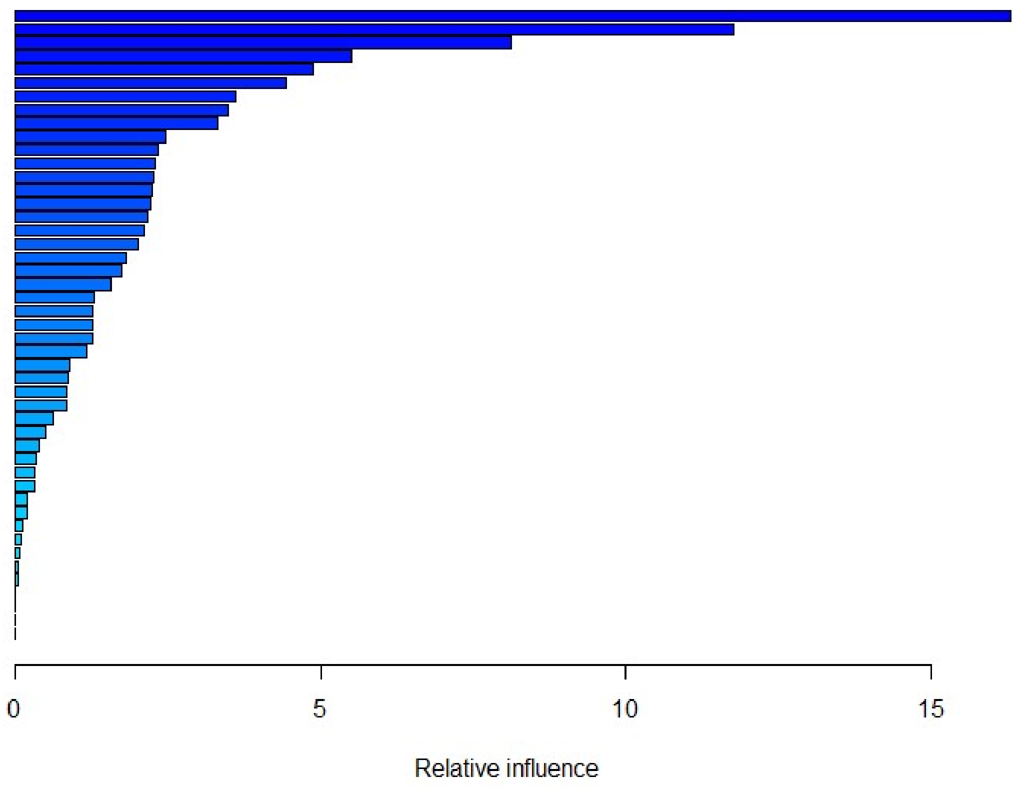
```
                                             var      rel.inf
CurrentEquipmentDays          CurrentEquipmentDays 16.12788581
MonthsInService                    MonthsInService 12.83316521
MonthlyMinutes                      MonthlyMinutes  8.05463713
MonthlyRevenue                      MonthlyRevenue  5.47555525
CreditRating                          CreditRating  5.30727996
TotalRecurringCharge          TotalRecurringCharge  4.37050439
OverageMinutes                      OverageMinutes  3.86227130
AgeHH1                                      AgeHH1  3.70958990
RoamingCalls                          RoamingCalls  3.40397648
OffPeakCallsInOut                OffPeakCallsInOut  3.14166537
HandsetPrice                          HandsetPrice  2.55222101
DroppedCalls                          DroppedCalls  2.39060628
InboundCalls                          InboundCalls  2.21126458
RetentionCalls                      RetentionCalls  1.93125146
OutboundCalls                        OutboundCalls  1.80763052
Occupation                              Occupation  1.63062825
DroppedBlockedCalls            DroppedBlockedCalls  1.60177785
```



```
ReceivedCalls                                  ReceivedCalls  1.56838802
UnansweredCalls                              UnansweredCalls  1.54195907
PeakCallsInOut                                PeakCallsInOut  1.50360215
BlockedCalls                                    BlockedCalls  1.39120927
HandsetWebCapable                          HandsetWebCapable  1.38268855
AgeHH2                                                AgeHH2  1.29050619
CustomerCareCalls                          CustomerCareCalls  1.25253809
HandsetRefurbished                        HandsetRefurbished  1.21400730
IncomeGroup                                      IncomeGroup  1.06417445
MadeCallToRetentionTeam              MadeCallToRetentionTeam  1.04567385
ActiveSubs                                        ActiveSubs  1.03461956
PrizmCode                                          PrizmCode  0.91775251
Handsets                                            Handsets  0.84458909
ThreewayCalls                                  ThreewayCalls  0.76079838
CallWaitingCalls                            CallWaitingCalls  0.52490412
HandsetModels                                  HandsetModels  0.50632684
RespondsToMailOffers                    RespondsToMailOffers  0.47395534
OwnsComputer                                    OwnsComputer  0.35742272
NewCellphoneUser                            NewCellphoneUser  0.27484219
```



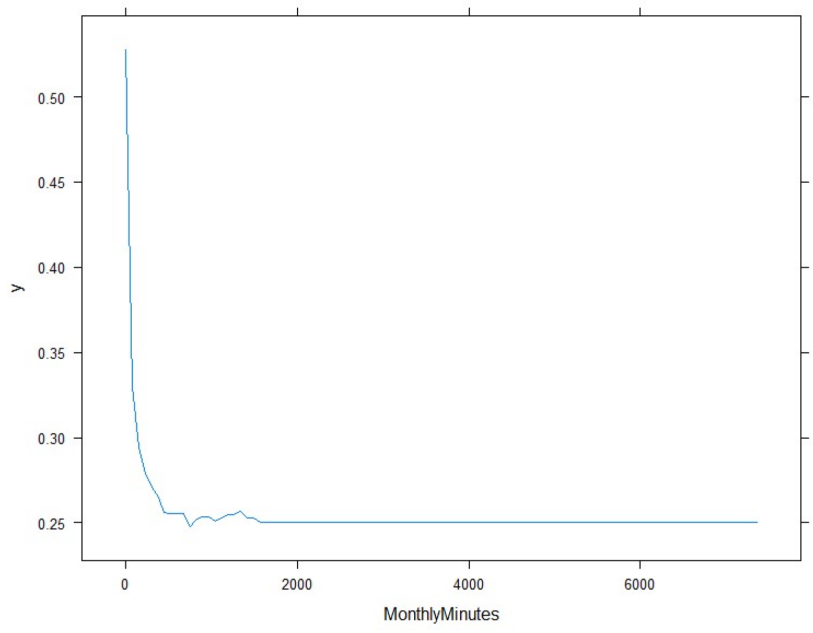
```
CallForwardingCalls                      CallForwardingCalls  0.21733126
OwnsMotorcycle                                OwnsMotorcycle  0.14491158
ChildrenInHH                                    ChildrenInHH  0.13099211
RetentionOffersAccepted              RetentionOffersAccepted  0.09170179
ReferralsMadeBySubscriber ReferralsMadeBySubscriber  0.05319483
TruckOwner                                        TruckOwner  0.00000000
RVOwner                                              RVOwner  0.00000000
BuysViaMailOrder                            BuysViaMailOrder  0.00000000
OptOutMailings                                OptOutMailings  0.00000000
HasCreditCard                                  HasCreditCard  0.00000000
MaritalStatus                                  MaritalStatus  0.00000000
```



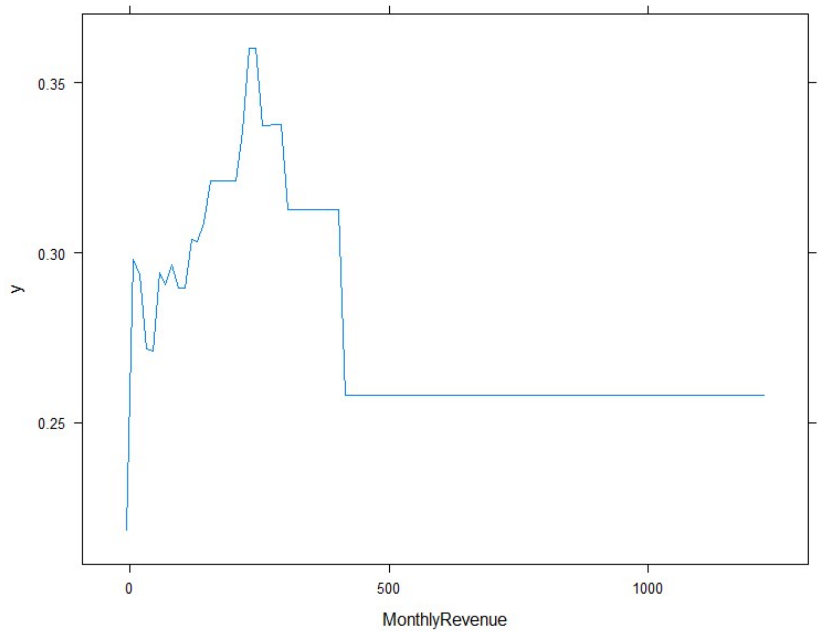
3 – Customer Churn Profile

3.1 – Marginal Predictor Plots

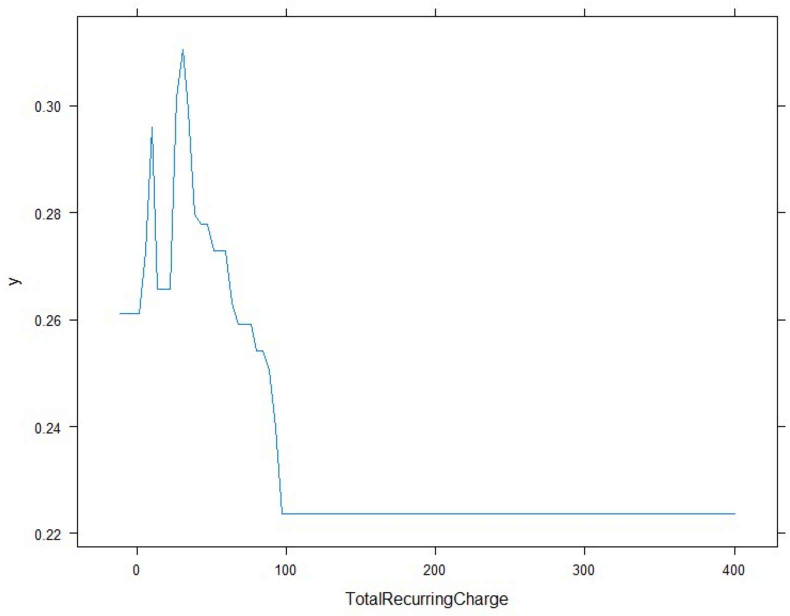
<u>MonthlyMinutes:</u>



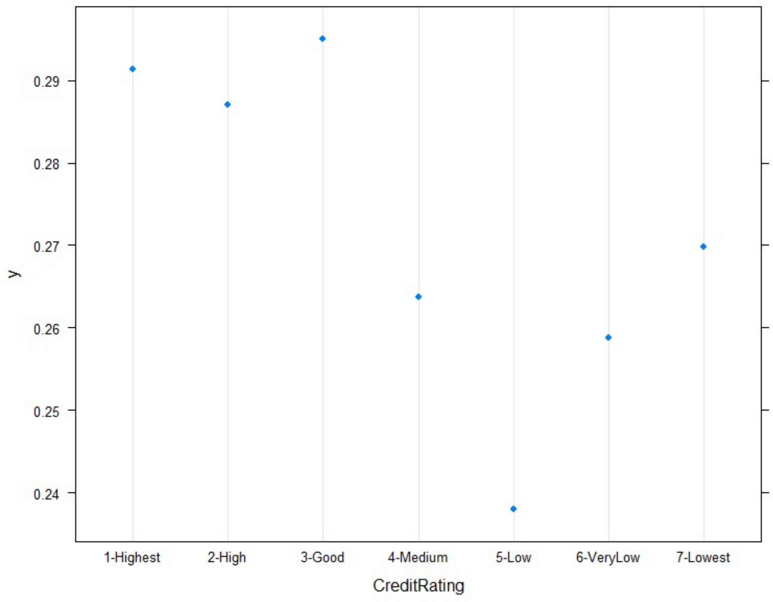
<u>MonthlyRevenue:</u>



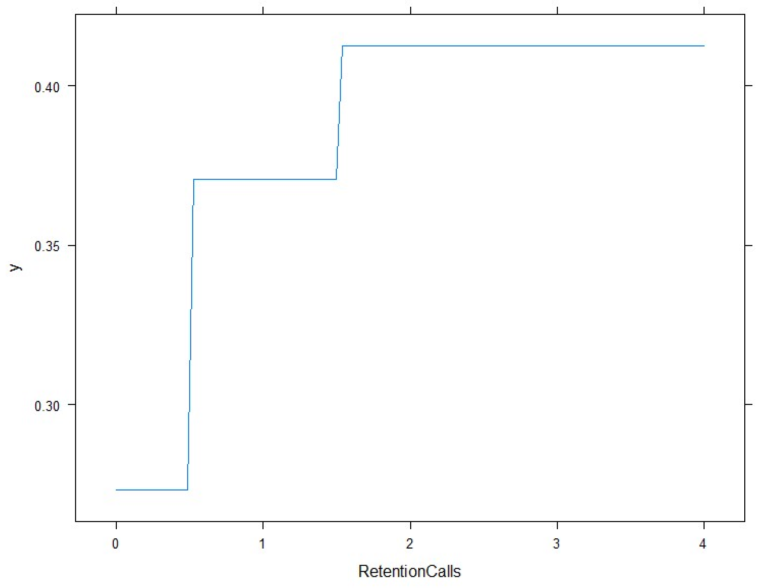
TotalRecurringCharge:



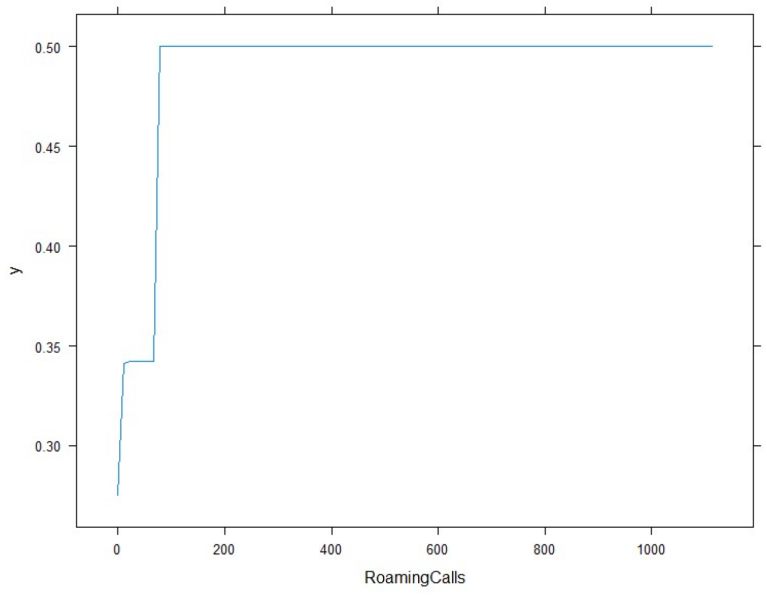
CreditRating:



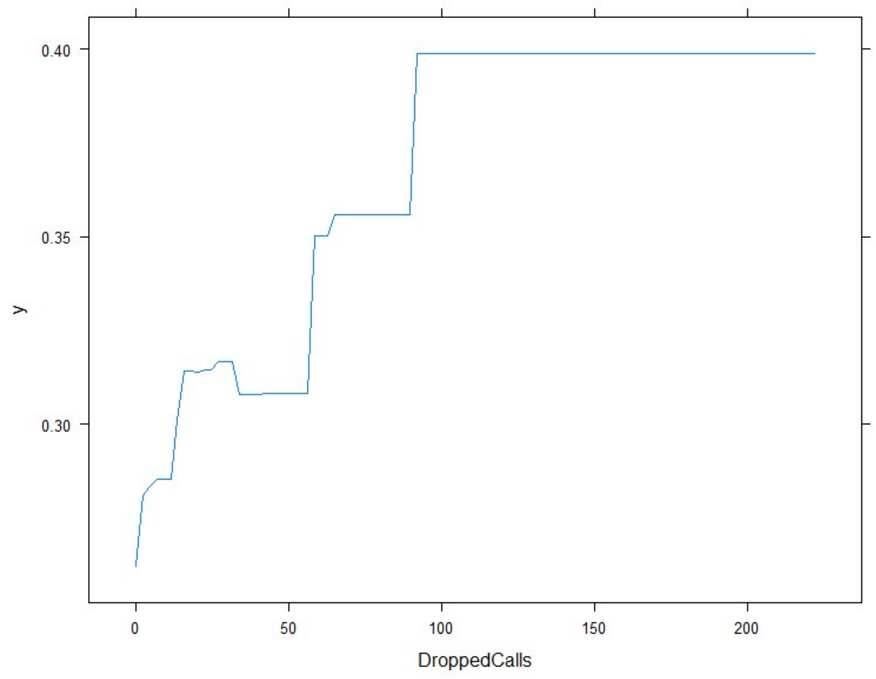
RetentionCalls:



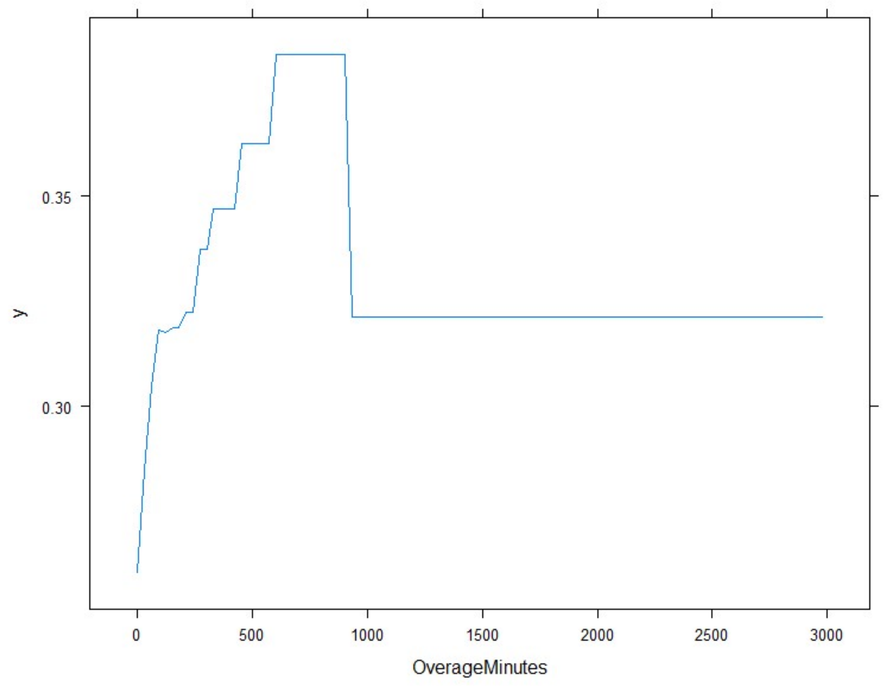
RoamingCalls:



<u>DroppedCalls:</u>



<u>OverageMinutes:</u>



3.2 – Customer Churn Profile Infographic (Attached Below)

CUSTOMER CHURN PROFILE

CurrentEquipmentDays
As current equipment gets older the likelihood of churn increases. Probability of churn is elevated past average at around 300 days as payment plans come to an end, and remain elevated and increasing beyond this.

AgeHH1
The older the first household member the less likely they are to churn, with lower than average starting from ~33yrs old. Therefore retention advertising should be focused on younger individuals.

MonthsInService
Likelihood of Churn spikes heavily towards the end of the first year of service as plans come to an end. If this hurdle is overcame the likelihood to churn decreases with time.

MonthlyMinutes/Calls
Higher usage customers, indicated by monthly minutes and all types of calls, are less likely to churn. Advertising should be focused to those who have less use of the service.

Dropped/RoamingCalls
Once dropped/roaming calls increase above 0 a customer is more likely than average to churn, with this likelihood increasing as these incidents increase. Investigation into why these scenarios are occurring and deals should be offered to those who experience them to ensure they stay until the issues are fixed.

RetentionCalls
The more retention calls are made, the higher the likelihood for a customer to churn. This process needs to be reviewed to determine how to more effectively ensure those who are thinking of churning are retained.

MonthlyRevenue/CreditRating
The higher the revenue the more likely a customer is to Churn. This is likely a combination of the regular plan pricing as well as overage/roaming minutes. Customers who generate the most revenue should be offered special deals to ensure they are retained and continue to generate revenue. Good-High credit rating customers are slightly more likely to churn while Medium-Lowest credit rating customers are least likely to churn. Focus advertising to higher credit rating customers as they are reliable income.