Home Work 3 – Problem (Selected Seed# 28)

For this homework we will use (a variant of) the churn dataset discussed in the Data Science for Business text.

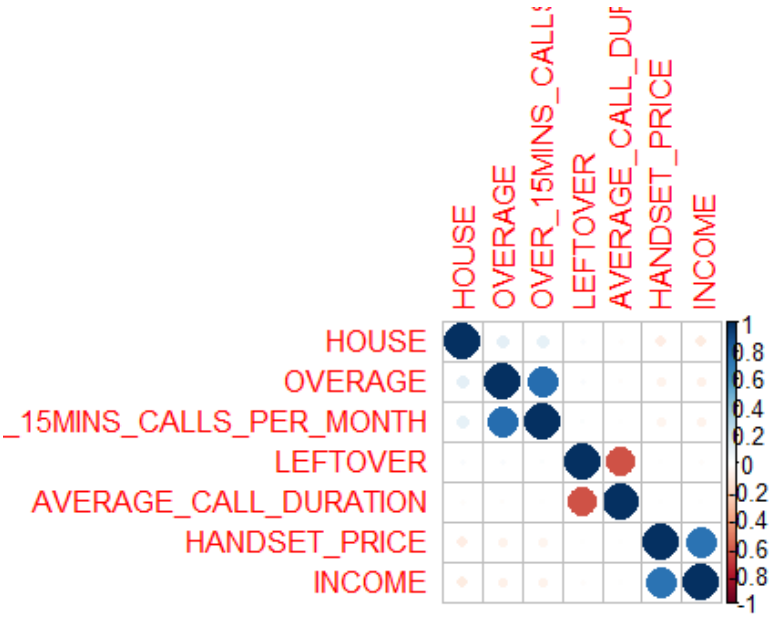
Begin by doing a brief exploratory analysis, as in the first homework. Review the univariate distributions, missing values, etc. and do any cleaning / transformations that you think might be useful. Details are not necessary here - the goal is to understand the data and properly prepare it for modeling.

Find the best model you can to predict the target variable "result". Consider all the different models we have examined in class, tuned for this problem. Use area under the ROC curve as your evaluation criterion. Briefly describe the process you used to choose the best model of each type - proper use of training/validation/testing sets, parameter tuning, etc. Show plots for the best model of each type (best decision tree, best SVM, etc.).

Finally, using your best overall model, score your test set and build a profit curve. Assume that the cost for an intervention (say, a discount given to a predicted churner) is $100, that all interventions are successful, and that retaining a customer is worth $1000. Based on this result, give a specific recommendation for how many people should be given the discount.

Exploration and Transformation

Made a quick correlation analysis and found a few correlations between the different data features. Made sure there weren't any missing values and all numeric values were transformed to the 0-1 scale. This was required for creating models based on neural network.



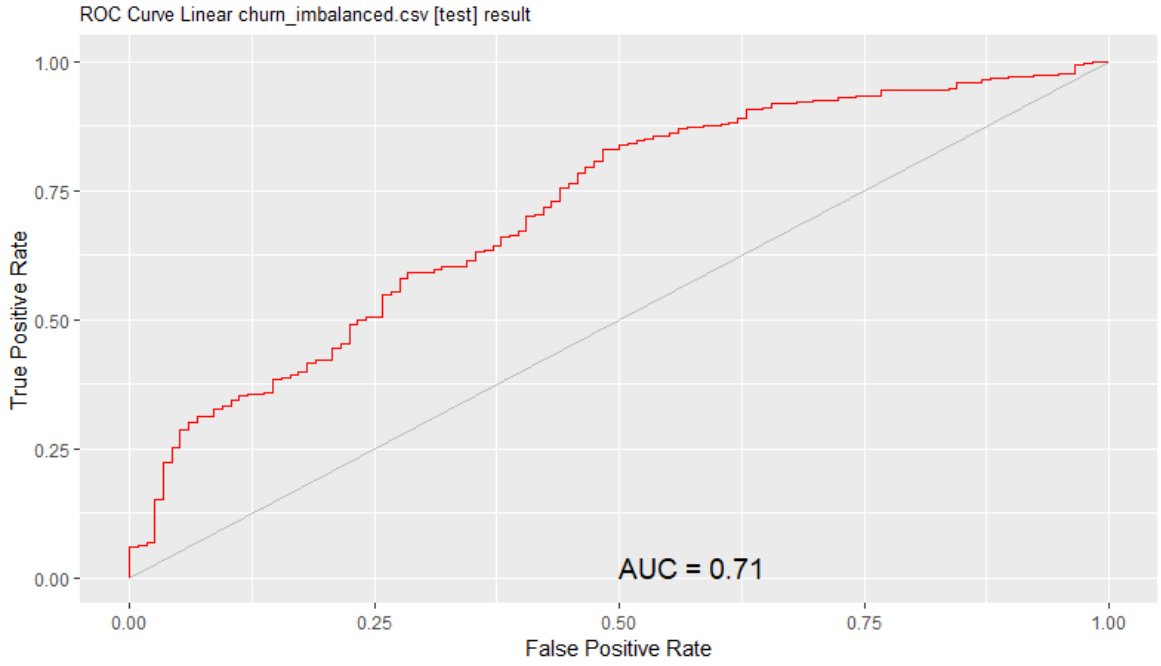
Linear Logistic Regression

Based on the logistic regression, identified the insignificant features and removed them so that a more accurate model can be made. It was found that the AUC got improved from 0.65 to 0.67 when these insignificant variables were removed.

```
NULL
COLLEGE                          0.913558
REPORTED_SATISFACTION            0.642602
REPORTED_USAGE_LEVEL             0.918857
CONSIDERING_CHANGE_OF_PLAN       0.424968
R01_INCOME                       0.001582 **
R01_OVERAGE                    < 2.2e-16 ***
R01_LEFTOVER                     0.070399 .
R01_HOUSE                      < 2.2e-16 ***
R01_HANDSET_PRICE                0.985917
R01_OVER_15MINS_CALLS_PER_MONTH  0.033147 *
R01_AVERAGE_CALL_DURATION        0.037124 *
```

Validation Data	AUC
With all features	0.65
Only with significant features	0.67

Test data AUC plot based on the best linear logistic model (only with significant features):

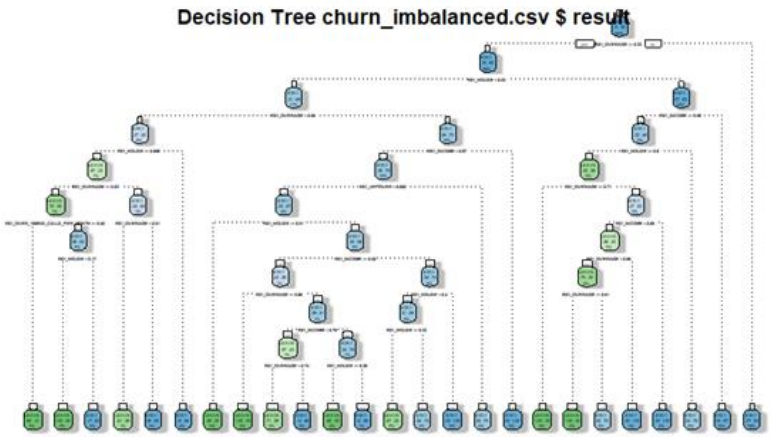


Confusion Matrix for the best-case scenario for both validation and test data (proportions)

Validation Data	Testing Data
Predicted Actual LEAVE STAY Error LEAVE 0 8.4 100 STAY 0 91.6 0 Overall error: 8.4%, Averaged class error: 50%	Predicted Actual LEAVE STAY Error LEAVE 0 10.2 100 STAY 0 89.8 0 Overall error: 10.2%, Averaged class error: 50%

Decision Tree

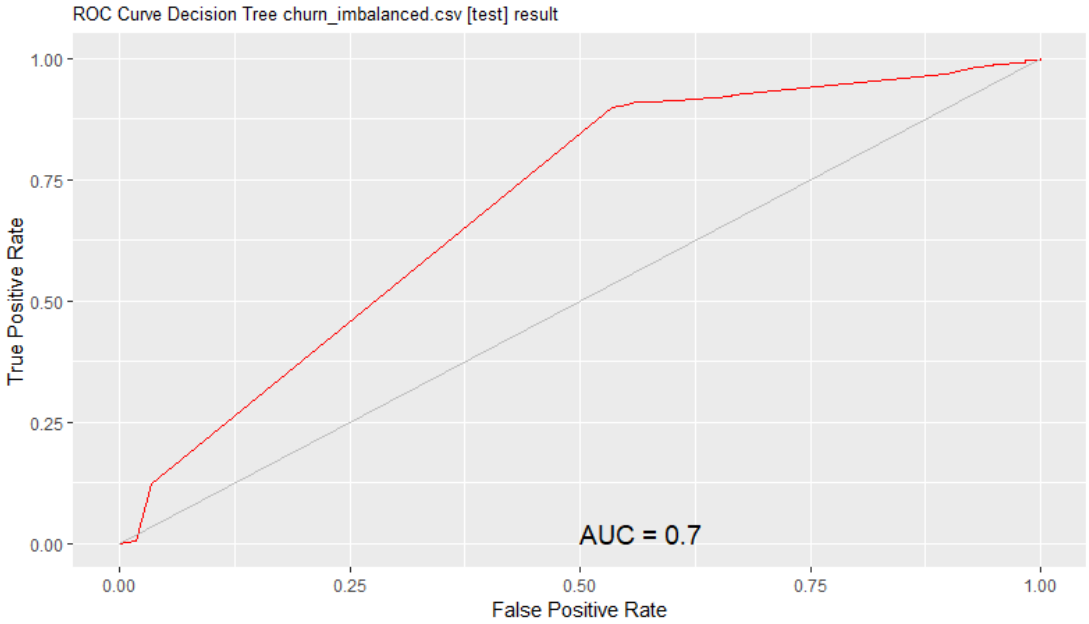
It was found that the xerror wasn't getting reduced. But when the tree was pruned at the right complexity level, it was found that the AUC getting improved. Area under the ROC curve was first found to be 0.56. But pruning the tree at complexity level, 0.003, found to improve the AUC to 0.69




```
Variables actually used in tree constru
 R01_HOUSE
 R01_LEFTOVER
 R01_OVERAGE
 R01_INCOME
 R01_OVER_15MINS_CALLS_PER_MONTH
```

Validation Data	AUC
Before pruning	0.56
After pruning	0.69

Test data AUC plot based on the best decision tree model (after pruning):



Confusion Matrix for the best-case scenario for both validation and test data (proportions)

Validation Data	Testing Data
Predicted Actual LEAVE STAY Error LEAVE 0.5 7.8 93.7 STAY 1.2 90.4 1.3 Overall error: 9.1%, Averaged class error: 47.5%	Predicted Actual LEAVE STAY Error LEAVE 0.7 9.5 93.1 STAY 1.7 88.1 1.9 Overall error: 11.2%, Averaged class error: 47.5%

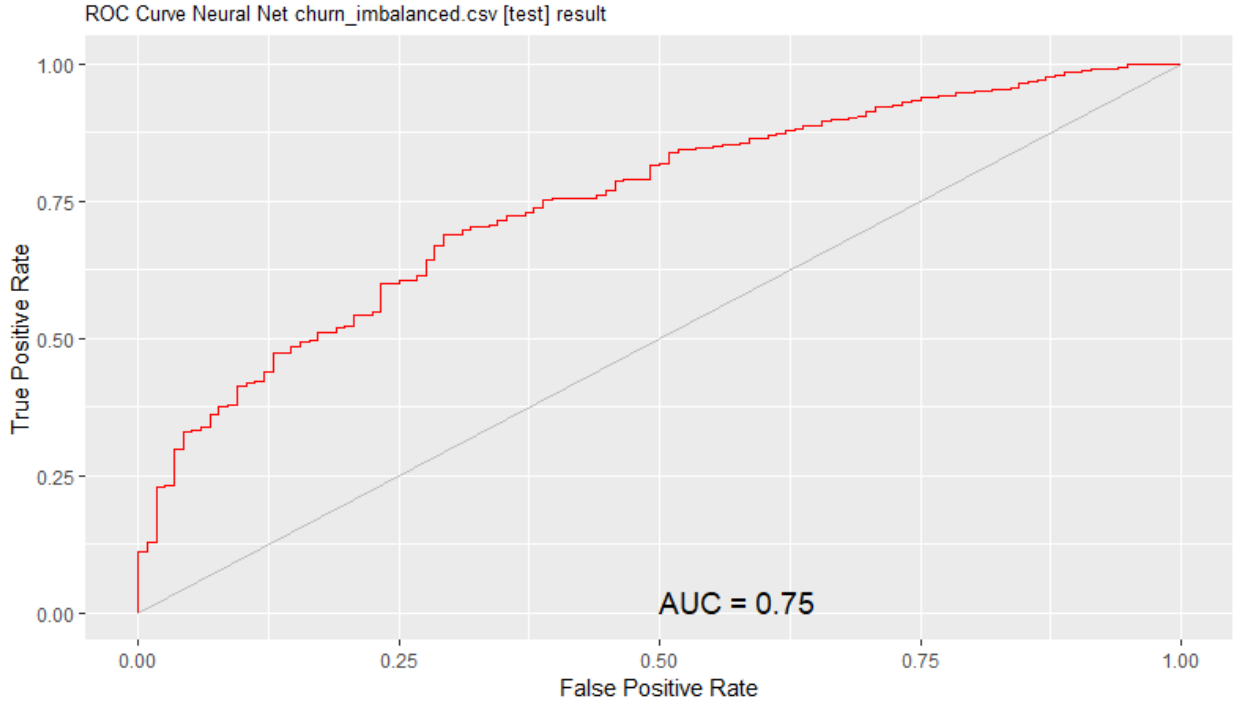
Neural Network

Neural network gave a better performance with 3 hidden layer nodes. The AUC found to be the best in this scenario which was found to be 0.71. AUC was compared starting from 10 hidden nodes and reducing it by 1 each time.

```
Summary of the Neural Net model (built using nnet):
A 6-3-1 network with 31 weights.
Inputs: R01_INCOME, R01_OVERAGE, R01_LEFTOVER, R01_HOUSE,
R01_OVER_15MINS_CALLS_PER_MONTH, R01_AVERAGE_CALL_DURATION.
Output: as.factor(result).
Sum of Squares Residuals: 423.3556.
```

Validation Data Hidden Layers	AUC
10	0.70
5	0.68
4	0.68
3	0.71
2	0.67

Test data AUC plot based on the best neural network model (hidden layers=3):



Confusion Matrix for the best-case scenario for both validation and test data (proportions)

Validation Data	Testing Data
Predicted Actual LEAVE STAY Error LEAVE 0 8.4 100 STAY 0 91.6 0 Overall error: 8.4%, Averaged class error: 50%	Predicted Actual LEAVE STAY Error LEAVE 0 10.2 100 STAY 0 89.8 0 Overall error: 10.2%, Averaged class error: 50%

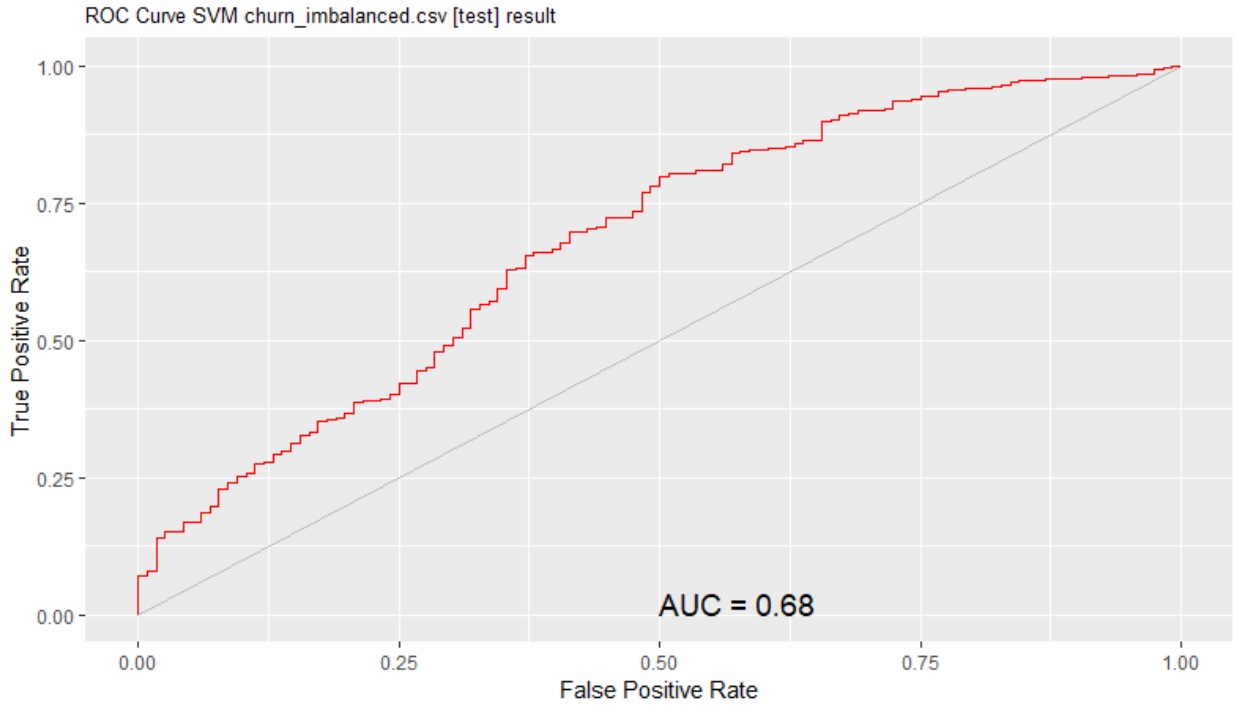
Support Vector Machine

Polynomial kernel function was used for creating the SVN model. The model with degree, one, gave the best AUC for this SVM model. The model was tried with degrees one, two and three.

```
Support Vector Machine object of class "ksvm"
SV type: C-svc  (classification)
 parameter : cost C = 1
Polynomial kernel function.
 Hyperparameters : degree =  1  scale =  1  offset =  1
Number of Support Vectors : 1073
Objective Function Value : -1018
Training error : 0.095911
Probability model included.
```

Validation Data Polynomial kernel degree	AUC
1 (1.45 secs)	0.65
2 (1.53 secs)	0.58
3 (16.02 mins)	0.47

Test data AUC plot based on the best SVN model (polynomial kernel degree=1):



Confusion Matrix for the best-case scenario for both validation and test data (proportions)

Validation Data	Testing Data
Predicted Actual LEAVE STAY Error LEAVE 0 8.4 100 STAY 0 91.6 0 Overall error: 8.4%, Averaged class error: 50%	Predicted Actual LEAVE STAY Error LEAVE 0 10.2 100 STAY 0 89.8 0 Overall error: 10.2%, Averaged class error: 50%

Ensemble Boosting (ada algorithm)

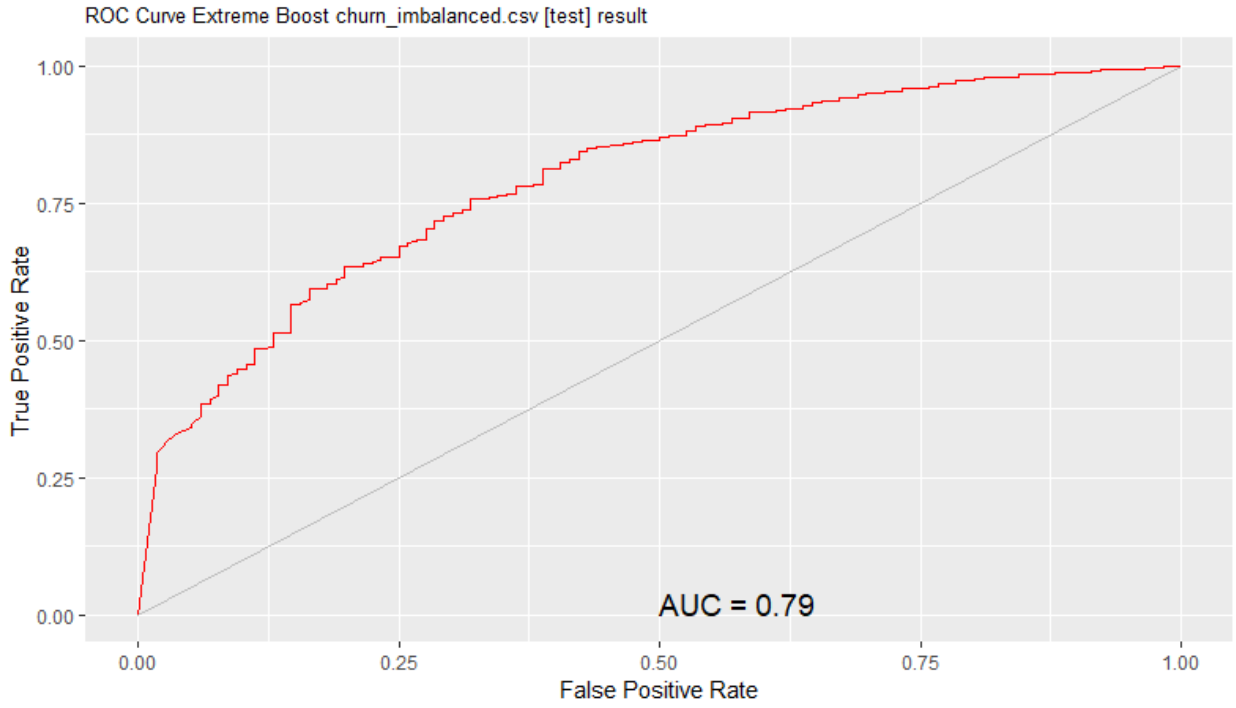
Used the Adaptive algorithm for creating the Boosting model. Several models were tried with number of trees to find the best model. The best AUC was found to be 0.75. Variables used for this model and their frequency is given below.

```
Frequency of variables actually used:

                                R01_HOUSE                                R01_INCOME
                                       34                                        32
                              R01_OVERAGE                              R01_LEFTOVER
                                       28                                        23
R01_OVER_15MINS_CALLS_PER_MONTH       18  R01_AVERAGE_CALL_DURATION              13
```

Validation Data # of Trees	AUC
30	0.748
35	0.750
40	0.753
45	0.751

Test data AUC plot based on the best Boosting model (number of trees=40):



Confusion Matrix for the best-case scenario for both validation and test data (proportions)

Validation Data	Testing Data
Predicted Actual LEAVE STAY Error LEAVE 0 8.4 100 STAY 0 91.6 0 Overall error: 8.4%, Averaged class error: 50%	Predicted Actual LEAVE STAY Error LEAVE 0.1 10.1 99.1 STAY 0.1 89.7 0.1 Overall error: 10.2%, Averaged class error: 49.6%

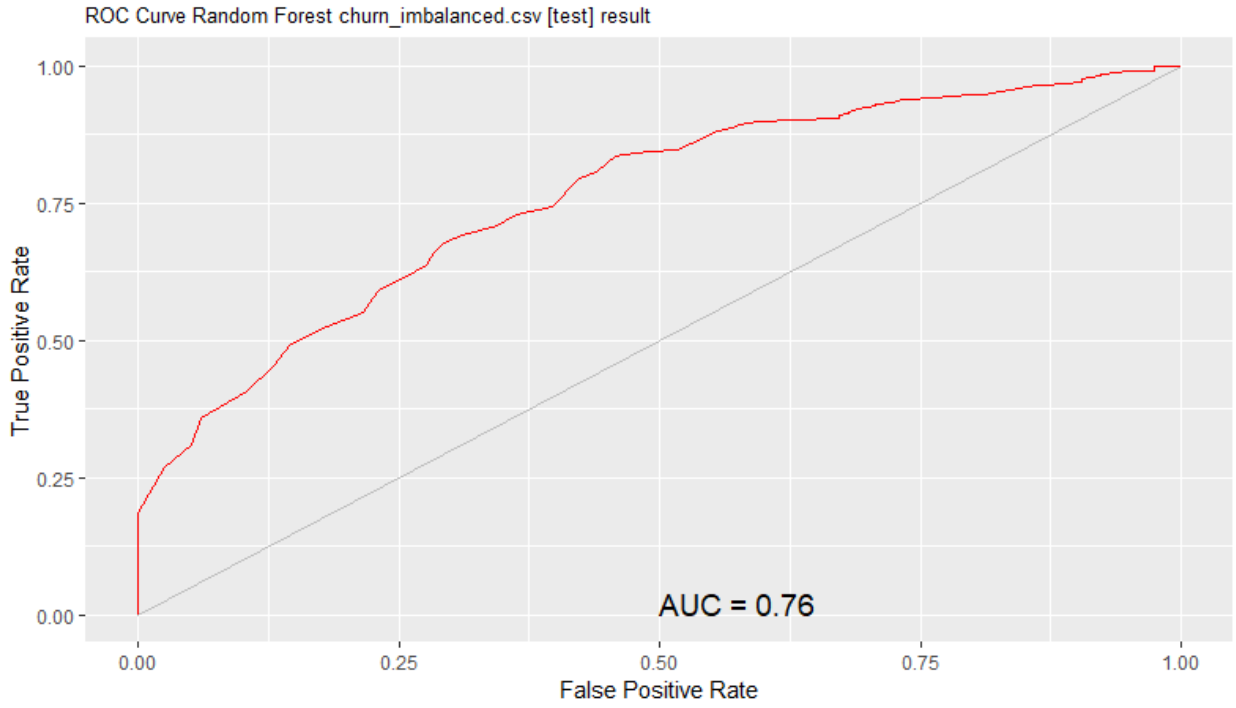
Random Forest

Used the traditional algorithm to create the random forest model. Number of trees were adjusted to fine tune the algorithm. Even though there wasn't a big difference between the different values, a better AUC value was obtained when the number of trees was 150.

```
Variable Importance
====================
                              MeanDecreaseAccuracy LEAVE  STAY
R01_HOUSE                                    23.46 14.25 21.00
R01_OVERAGE                                  21.93 11.15 17.83
R01_INCOME                                   12.67  4.48 11.60
R01_OVER_15MINS_CALLS_PER_MONTH              12.62  3.40 10.74
R01_LEFTOVER                                 10.27 -2.56 11.38
R01_AVERAGE_CALL_DURATION                     8.40 -4.65  9.43
```

Validation Data # of Trees	AUC
500	0.712
250	0.714
125	0.712
200	0.714
150	0.716
175	0.714

Test data AUC plot based on the best Random Forest model (number of trees=150):



Confusion Matrix for the best-case scenario for both validation and test data (proportions)

Validation Data	Testing Data
Predicted Actual LEAVE STAY Error LEAVE 0.2 8.2 97.9 STAY 0.7 90.9 0.8 Overall error: 8.9%, Averaged class error: 49.35%	Predicted Actual LEAVE STAY Error LEAVE 0.3 9.9 97.4 STAY 0.7 89.1 0.8 Overall error: 10.6%, Averaged class error: 49.1%

Evaluation of Model performance

Based on the performance of each of the models with the test dataset, it was found that the Ensemble Booting Model was the best model of all. The below table shows a comparison of the best AUC's between the different models that were used in this assignment.

Model Name	AUC - Validation dataset	AUC - Testing dataset
Logistic Linear Model	0.67	0.71
Decision Tree	0.69	0.70
SVM	0.65	0.68
Neural Network	0.71	0.75
Random Forest	0.72	0.76
Boosting	0.75	0.79

Profit Curve

Scoring of the test dataset was done based on Boosting model to generate the profit curve. Cost for intervention on each churner was $100. But retaining a customer would worth $1000. Profit for each retained customer would then be $900. Profit calculation equation was

```
=IF(A2="STAY",D1-100,D1+900)
```

Excel spreadsheet row with maximum profit was found to be at row, 199. Total number of datasets in the spreadsheet was 1138. Based on the test data, maximum profit can be achieved if 199 (17.48%) people were given the discount. The profit curve based on the above profit calculation can be found below:

198	LEAVE	0.866618	0.133382	25300
199	LEAVE	0.866786	0.133214	26200
200	STAY	0.867422	0.132578	26100