

Business Analytics-Group Project Accuracy check

Khushboo Yadav

11/24/2020

Conclusion : As per the result decision tree seems to be more accurate .

```
library(plyr)
library(ggplot2)
library(caret)
```

1.Import Libraries

```
## Loading required package: lattice
```

```
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
##
```

```
## Attaching package: 'modeltools'
```

```
## The following object is masked from 'package:plyr':
```

```
##
```

```
##      empty
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
library(rpart)
```

```
library(rpart.plot)
```

```
Churn_Train <- read.csv("Churn_Train(1).csv")
summary(Churn_Train)
```

2.Reading the dataset

```

##      state      account_length      area_code      international_plan
## Length:3333      Min.      :-209.00      Length:3333      Length:3333
## Class :character      1st Qu.: 72.00      Class :character      Class :character
## Mode  :character      Median : 100.00      Mode  :character      Mode  :character
##                               Mean  : 97.32
##                               3rd Qu.: 127.00
##                               Max.   : 243.00
##                               NA's    :501
## voice_mail_plan      number_vmail_messages      total_day_minutes      total_day_calls
## Length:3333      Min.      :-10.000      Min.      : 0.0      Min.      : 0.0
## Class :character      1st Qu.: 0.000      1st Qu.: 149.3      1st Qu.: 87.0
## Mode  :character      Median : 0.000      Median : 190.5      Median :101.0
##                               Mean   : 7.333      Mean   : 418.9      Mean   :100.3
##                               3rd Qu.: 16.000      3rd Qu.: 237.8      3rd Qu.:114.0
##                               Max.    : 51.000      Max.    :2185.1      Max.    :165.0
##                               NA's     :200      NA's     :200      NA's     :200
## total_day_charge      total_eve_minutes      total_eve_calls      total_eve_charge
## Min.      : 0.00      Min.      : 0.0      Min.      : 0.0      Min.      : 0.00
## 1st Qu.:24.45      1st Qu.: 170.5      1st Qu.: 87.0      1st Qu.:14.14
## Median :30.65      Median : 209.9      Median :100.0      Median :17.09
## Mean   :30.63      Mean   : 324.3      Mean   :100.1      Mean   :17.08
## 3rd Qu.:36.84      3rd Qu.: 257.6      3rd Qu.:114.0      3rd Qu.:20.00
## Max.    :59.64      Max.    :1244.2      Max.    :170.0      Max.    :30.91
## NA's     :200      NA's     :301      NA's     :200      NA's     :200
## total_night_minutes      total_night_calls      total_night_charge      total_intl_minutes
## Min.      : 23.2      Min.      : 33.0      Min.      : 1.040      Min.      : 0.00
## 1st Qu.:167.3      1st Qu.: 87.0      1st Qu.: 7.530      1st Qu.: 8.50
## Median :201.4      Median :100.0      Median : 9.060      Median :10.30
## Mean   :201.2      Mean   :100.1      Mean   : 9.054      Mean   :10.23
## 3rd Qu.:235.3      3rd Qu.:113.0      3rd Qu.:10.590      3rd Qu.:12.10
## Max.    :395.0      Max.    :175.0      Max.    :17.770      Max.    :20.00
## NA's     :200      NA's     :200      NA's     :200      NA's     :200
## total_intl_calls      total_intl_charge      number_customer_service_calls
## Min.      : 0.00      Min.      :0.000      Min.      :0.000
## 1st Qu.: 3.00      1st Qu.:2.300      1st Qu.:1.000
## Median : 4.00      Median :2.780      Median :1.000
## Mean   : 4.47      Mean   :2.762      Mean   :1.561
## 3rd Qu.: 6.00      3rd Qu.:3.270      3rd Qu.:2.000
## Max.    :20.00      Max.    :5.400      Max.    :9.000
## NA's     :301      NA's     :200      NA's     :200
## churn
## Length:3333
## Class :character
## Mode  :character
##
##
##
##

```

analysing count of NA value in the dataset

```
sapply(Churn_Train, function(x) sum(is.na(x))) # NA data
```

```
##           state          account_length
##           0              501
##           area_code      international_plan
##           0              0
##           voice_mail_plan  number_vmail_messages
##           0              200
##           total_day_minutes  total_day_calls
##           200             200
##           total_day_charge    total_eve_minutes
##           200             301
##           total_eve_calls      total_eve_charge
##           200             200
##           total_night_minutes  total_night_calls
##           200              0
##           total_night_charge    total_intl_minutes
##           200             200
##           total_intl_calls      total_intl_charge
##           301             200
## number_customer_service_calls    churn
##           200              0
```

NA values

```
## 'data.frame': 3333 obs. of 20 variables:
## $ state : chr "NV" "HI" "DC" "HI" ...
## $ account_length : int 125 108 82 NA 83 89 135 28 86 65 ...
## $ area_code : chr "area_code_510" "area_code_415" "area_code_415" "area_code_40" ...
## $ international_plan : chr "no" "no" "no" "no" ...
## $ voice_mail_plan : chr "no" "no" "no" "yes" ...
## $ number_vmail_messages : int 0 0 0 30 0 0 0 0 0 0 ...
## $ total_day_minutes : num 2013 292 300 110 337 ...
## $ total_day_calls : int 99 99 109 71 120 81 81 87 115 137 ...
## $ total_day_charge : num 28.7 49.6 51 18.8 57.4 ...
## $ total_eve_minutes : num 1108 221 181 182 227 ...
## $ total_eve_calls : int 107 93 100 108 116 74 114 92 112 83 ...
## $ total_eve_charge : num 14.9 18.8 15.4 15.5 19.3 ...
## $ total_night_minutes : num 243 229 270 184 154 ...
## $ total_night_calls : int 92 110 73 88 114 120 82 112 95 111 ...
## $ total_night_charge : num 10.95 10.31 12.15 8.27 6.93 ...
## $ total_intl_minutes : num 10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...
## $ total_intl_calls : int 7 9 4 8 7 4 6 3 7 6 ...
## $ total_intl_charge : num 2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...
## $ number_customer_service_calls: int 0 2 0 2 0 1 1 3 2 4 ...
## $ churn : chr "no" "yes" "yes" "no" ...

##
## Attaching package: 'mice'

## The following objects are masked from 'package:base':
##
## cbind, rbind
##
## iter imp variable
## 1 1 account_length number_vmail_messages total_day_minutes total_day_calls total_day_charge
## 1 2 account_length number_vmail_messages total_day_minutes total_day_calls total_day_charge
```

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning: Number of logged events: 7
```

##	state	account_length
##	0.000000	0.000000
##	area_code	international_plan
##	0.000000	0.000000
##	voice_mail_plan	number_vmail_messages
##	0.000000	0.000000
##	total_day_minutes	total_day_calls
##	0.000000	0.000000
##	total_day_charge	total_eve_minutes
##	0.000000	0.000000
##	total_eve_calls	total_eve_charge
##	0.000000	0.000000
##	total_night_minutes	total_night_calls
##	0.000000	0.000000
##	total_night_charge	total_intl_minutes
##	0.060006	0.000000

```
##          total_intl_calls          total_intl_charge
##          0.000000          0.060006
## number_customer_service_calls          churn
##          0.000000          0.000000
```

data manipulation ##updating the values of International plan , voice mail plan and churn to 1 or 0

```
#for Churn_Train data
Churn_Train$international_plan<-ifelse(Churn_Train$international_plan=="yes",1,0)
Churn_Train$voice_mail_plan<- ifelse(Churn_Train$voice_mail_plan=="yes",1,0)
Churn_Train$churn<- ifelse(Churn_Train$churn=="yes",1,0)
##Factorization of above data
#for Churn_Train data
Churn_Train$international_plan<-as.factor(Churn_Train$international_plan)
Churn_Train$voice_mail_plan <-as.factor(Churn_Train$voice_mail_plan)
Churn_Train$churn<- as.factor(Churn_Train$churn)
Churn_Train$area_code<- as.factor(Churn_Train$area_code) # added because of decision trees
Churn_Train$state<- as.factor(Churn_Train$state)
summary(Churn_Train)
```

```
##      state      account_length      area_code      international_plan
## WV       : 106      Min.      :-209.00      area_code_408: 838      0:3010
## MN       :  84      1st Qu.:  71.00      area_code_415:1655      1: 323
## NY       :  83      Median : 100.00      area_code_510: 840
## AL       :  80      Mean       :  97.09
## OH       :  78      3rd Qu.: 127.00
## OR       :  78      Max.       : 243.00
## (Other):2824
## voice_mail_plan number_vmail_messages total_day_minutes total_day_calls
## 0:2411          Min.      :-10.000      Min.       :  0.0      Min.       :  0.0
## 1: 922          1st Qu.:  0.000      1st Qu.: 147.0      1st Qu.:  87.0
##          Median :  0.000      Median : 191.0      Median :101.0
##          Mean    :  7.331      Mean    : 418.0      Mean    :100.2
##          3rd Qu.: 16.000      3rd Qu.: 242.6      3rd Qu.:114.0
##          Max.    : 51.000      Max.    :2185.1      Max.    :165.0
##
## total_day_charge total_eve_minutes total_eve_calls total_eve_charge
## Min.      : 0.00      Min.       :  0.0      Min.       :  0.0      Min.       :  0.00
## 1st Qu.:24.51      1st Qu.: 168.7      1st Qu.:  87.0      1st Qu.:14.20
## Median :30.60      Median : 209.9      Median :100.0      Median :17.09
## Mean    :30.58      Mean    : 319.7      Mean    :100.1      Mean    :17.07
## 3rd Qu.:36.70      3rd Qu.: 258.4      3rd Qu.:114.0      3rd Qu.:19.92
## Max.    :59.64      Max.    :1244.2      Max.    :170.0      Max.    :30.91
##
## total_night_minutes total_night_calls total_night_charge total_intl_minutes
## Min.      : 23.2      Min.       : 33.0      Min.       : 1.040      Min.       :  0.00
## 1st Qu.:167.8      1st Qu.:  87.0      1st Qu.:  7.530      1st Qu.:  8.50
## Median :202.0      Median :100.0      Median :  9.060      Median :10.30
## Mean    :201.5      Mean    :100.1      Mean    :  9.054      Mean    :10.22
## 3rd Qu.:235.8      3rd Qu.:113.0      3rd Qu.:10.590      3rd Qu.:12.10
## Max.    :395.0      Max.    :175.0      Max.    :17.770      Max.    :20.00
##
##          NA's      :200
## total_intl_calls total_intl_charge number_customer_service_calls churn
## Min.      : 0.000      Min.       :0.000      Min.       :0.000          0:2850
## 1st Qu.:  3.000      1st Qu.:2.300      1st Qu.:1.000          1: 483
```



```
## Median : 4.000    Median :2.780    Median :1.000
## Mean   : 4.488    Mean   :2.762    Mean   :1.561
## 3rd Qu.: 6.000    3rd Qu.:3.270    3rd Qu.:2.000
## Max.   :20.000    Max.   :5.400    Max.   :9.000
##                               NA's   :200
```

```
str(Churn_Train)
```

```
## 'data.frame':   3333 obs. of  20 variables:
## $ state          : Factor w/ 51 levels "AK","AL","AR",...: 34 12 8 12 36 25 28 39 13 1
## $ account_length : int  125 108 82 31 83 89 135 28 86 65 ...
## $ area_code      : Factor w/ 3 levels "area_code_408",...: 3 2 2 1 2 2 2 1 2 ...
## $ international_plan : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ...
## $ voice_mail_plan  : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 ...
## $ number_vmail_messages : int  0 0 0 30 0 0 0 0 0 0 ...
## $ total_day_minutes : num  2013 292 300 110 337 ...
## $ total_day_calls   : int  99 99 109 71 120 81 81 87 115 137 ...
## $ total_day_charge  : num  28.7 49.6 51 18.8 57.4 ...
## $ total_eve_minutes : num  1108 221 181 182 227 ...
## $ total_eve_calls   : int  107 93 100 108 116 74 114 92 112 83 ...
## $ total_eve_charge  : num  14.9 18.8 15.4 15.5 19.3 ...
## $ total_night_minutes : num  243 229 270 184 154 ...
## $ total_night_calls : int  92 110 73 88 114 120 82 112 95 111 ...
## $ total_night_charge : num  10.95 10.31 12.15 8.27 6.93 ...
## $ total_intl_minutes : num  10.9 14 11.7 11 15.8 9.1 10.3 10.1 9.8 12.7 ...
## $ total_intl_calls   : int  7 9 4 8 7 4 6 3 7 6 ...
## $ total_intl_charge  : num  2.94 3.78 3.16 2.97 4.27 2.46 2.78 2.73 2.65 3.43 ...
## $ number_customer_service_calls: int  0 2 0 2 0 1 1 3 2 4 ...
## $ churn              : Factor w/ 2 levels "0","1": 1 2 2 1 2 1 1 1 1 2 ...
```

```
##Churn Train data partitioning (60%,40%)
```

```
set.seed(2020)
```

```
partition<- createDataPartition(Churn_Train$churn,p=0.6,list=FALSE)
```

```
train_data<- Churn_Train[partition,]
```

```
validation_data<- Churn_Train[-partition,]
```

Accuracy for logistic regression

```
Model_Train <- glm(churn ~ .,family=binomial(link="logit"),data=train_data)
```

```
summary(Model_Train)
```

```
##
## Call:
## glm(formula = churn ~ ., family = binomial(link = "logit"), data = train_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0603  -0.4905  -0.2921  -0.1549   3.0545
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.794e+00  1.318e+00  -7.428 1.10e-13 ***
## stateAL       6.401e-01  9.668e-01   0.662  0.50796
## stateAR       9.920e-01  9.574e-01   1.036  0.30015
```

## stateAZ	6.517e-01	1.034e+00	0.630	0.52853
## stateCA	2.026e+00	9.937e-01	2.039	0.04148 *
## stateCO	3.609e-01	9.946e-01	0.363	0.71670
## stateCT	8.115e-01	9.458e-01	0.858	0.39087
## stateDC	8.501e-01	1.034e+00	0.822	0.41082
## stateDE	1.324e+00	9.319e-01	1.421	0.15541
## stateFL	-2.484e-01	1.161e+00	-0.214	0.83058
## stateGA	8.249e-01	1.028e+00	0.802	0.42246
## stateHI	-2.519e-02	1.141e+00	-0.022	0.98239
## stateIA	2.923e-01	1.131e+00	0.258	0.79613
## stateID	9.516e-01	9.473e-01	1.004	0.31514
## stateIL	5.253e-01	1.069e+00	0.492	0.62303
## stateIN	1.048e-01	9.801e-01	0.107	0.91488
## stateKS	9.906e-01	9.962e-01	0.994	0.32005
## stateKY	7.978e-01	9.625e-01	0.829	0.40720
## stateLA	7.566e-01	1.032e+00	0.733	0.46351
## stateMA	9.812e-01	9.943e-01	0.987	0.32375
## stateMD	1.299e+00	9.276e-01	1.400	0.16138
## stateME	1.252e+00	9.704e-01	1.290	0.19697
## stateMI	1.090e+00	9.614e-01	1.134	0.25672
## stateMN	1.410e+00	9.090e-01	1.552	0.12072
## stateMO	3.471e-01	1.061e+00	0.327	0.74366
## stateMS	1.534e+00	9.544e-01	1.607	0.10795
## stateMT	1.990e+00	9.182e-01	2.168	0.03018 *
## stateNC	3.070e-01	1.023e+00	0.300	0.76397
## stateND	8.085e-01	9.797e-01	0.825	0.40923
## stateNE	7.837e-01	9.927e-01	0.789	0.42986
## stateNH	1.315e+00	9.612e-01	1.368	0.17120
## stateNJ	1.788e+00	9.228e-01	1.938	0.05262 .
## stateNM	3.859e-01	1.004e+00	0.385	0.70061
## stateNV	1.554e+00	9.196e-01	1.690	0.09107 .
## stateNY	9.468e-01	9.488e-01	0.998	0.31836
## stateOH	-3.616e-01	1.138e+00	-0.318	0.75057
## stateOK	2.362e-01	1.044e+00	0.226	0.82101
## stateOR	-1.898e-01	1.088e+00	-0.174	0.86154
## statePA	1.494e+00	9.669e-01	1.546	0.12222
## stateRI	-3.172e-02	1.040e+00	-0.030	0.97567
## stateSC	1.993e+00	9.617e-01	2.072	0.03823 *
## stateSD	8.213e-01	1.003e+00	0.819	0.41298
## stateTN	1.051e+00	9.935e-01	1.058	0.29026
## stateTX	1.370e+00	9.184e-01	1.492	0.13574
## stateUT	1.684e+00	9.087e-01	1.854	0.06381 .
## stateVA	-3.109e-01	1.121e+00	-0.277	0.78149
## stateVT	-5.575e-01	1.238e+00	-0.451	0.65233
## stateWA	1.515e+00	9.195e-01	1.648	0.09936 .
## stateWI	-4.747e-01	1.061e+00	-0.448	0.65449
## stateWV	1.092e+00	9.165e-01	1.191	0.23358
## stateWY	3.740e-01	9.769e-01	0.383	0.70182
## account_length	9.425e-04	1.650e-03	0.571	0.56792
## area_codearea_code_415	1.694e-01	1.977e-01	0.857	0.39150
## area_codearea_code_510	-1.086e-01	2.319e-01	-0.468	0.63954
## international_plan1	2.330e+00	2.123e-01	10.975	< 2e-16 ***
## voice_mail_plan1	-1.339e+00	4.956e-01	-2.702	0.00688 **
## number_vmail_messages	1.156e-02	1.596e-02	0.724	0.46897

```

## total_day_minutes      -2.200e-03  2.784e-03  -0.790  0.42941
## total_day_calls        3.450e-03  3.998e-03   0.863  0.38811
## total_day_charge       9.478e-02  1.637e-02   5.790  7.05e-09 ***
## total_eve_minutes      4.326e-03  5.500e-03   0.787  0.43149
## total_eve_calls       -3.289e-03  3.942e-03  -0.834  0.40408
## total_eve_charge       5.084e-02  6.672e-02   0.762  0.44607
## total_night_minutes    6.066e-01  1.211e+00   0.501  0.61643
## total_night_calls      5.963e-04  4.102e-03   0.145  0.88442
## total_night_charge    -1.341e+01  2.691e+01  -0.498  0.61831
## total_intl_minutes     3.009e-01  7.519e+00   0.040  0.96808
## total_intl_calls      -7.914e-02  3.581e-02  -2.210  0.02709 *
## total_intl_charge     -7.529e-01  2.785e+01  -0.027  0.97843
## number_customer_service_calls  5.296e-01  5.757e-02   9.200  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 1561.4  on 1879  degrees of freedom
## Residual deviance: 1142.1  on 1810  degrees of freedom
##    (120 observations deleted due to missingness)
## AIC: 1282.1
##
## Number of Fisher Scoring iterations: 6

predict_validation<-predict(Model_Train,newdata = validation_data,type='response')
resultcheck<-ifelse(predict_validation>0.5,1,0)
## Accuracy check
error<-mean(resultcheck!=validation_data$churn)
accuracy<-1-error
print(accuracy)

## [1] NA

#
table(validation_data$churn, resultcheck > 0.5)

##
##      FALSE TRUE
## 0    1032   38
## 1     136   47

#confusion matrix
resultcheck<- as.factor(resultcheck)
confusionMatrix(resultcheck,validation_data$churn)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction    0      1
##      0    1032   136
##      1     38    47
##
##              Accuracy : 0.8611
##              95% CI : (0.8407, 0.8798)
##              No Information Rate : 0.854

```

```
##      P-Value [Acc > NIR] : 0.2499
##
##              Kappa : 0.2845
##
## Mcnemar's Test P-Value : 1.93e-13
##
##      Sensitivity : 0.9645
##      Specificity : 0.2568
##      Pos Pred Value : 0.8836
##      Neg Pred Value : 0.5529
##      Prevalence : 0.8540
##      Detection Rate : 0.8236
##      Detection Prevalence : 0.9322
##      Balanced Accuracy : 0.6107
##
##      'Positive' Class : 0
##
```

ROC for logistic regression

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
```

```
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      cov, smooth, var
```

```
#ROC Curve for validation Data set
```

```
roc(validation_data$churn, predict_validation)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
##
```

```
## Call:
```

```
## roc.default(response = validation_data$churn, predictor = predict_validation)
```

```
##
```

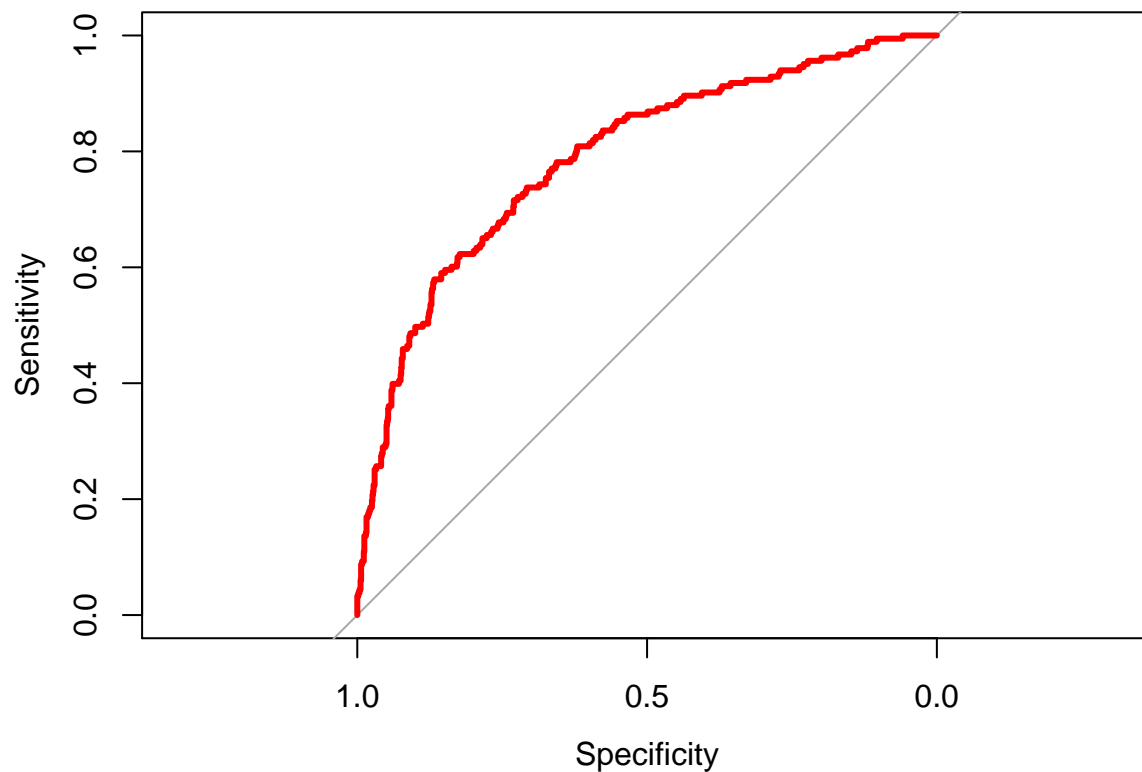
```
## Data: predict_validation in 1070 controls (validation_data$churn 0) < 183 cases (validation_data$churn 1)
```

```
## Area under the curve: 0.7892
```

```
plot.roc(validation_data$churn,predict_validation,col = "red", lwd = 3)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



Accuracy for decision tree

```
D_model <- ctree(churn~ ., train_data)
pred_tree <- predict(D_model, validation_data)
#table
table(pred_tree)
```

```
## pred_tree
##    0    1
## 1190  143
```

```
#confusion matrix
confusionMatrix(pred_tree, validation_data$churn)
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
## Prediction    0    1
##           0 1124   66
##           1   16  127
```

```
##
```

```
##           Accuracy : 0.9385
##           95% CI : (0.9242, 0.9508)
##    No Information Rate : 0.8552
##    P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 0.7216
```

```
##
```

```
##    McNemar's Test P-Value : 6.262e-08
```

```
##
##          Sensitivity : 0.9860
##          Specificity : 0.6580
##          Pos Pred Value : 0.9445
##          Neg Pred Value : 0.8881
##          Prevalence : 0.8552
##          Detection Rate : 0.8432
##          Detection Prevalence : 0.8927
##          Balanced Accuracy : 0.8220
##
##          'Positive' Class : 0
##
```

ROC for decision tree

```
pred_tree1 <- predict(D_model, validation_data, type='node')
#ROC Curve for validation Data set
roc(validation_data$churn, pred_tree1)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
##
```

```
## Call:
```

```
## roc.default(response = validation_data$churn, predictor = pred_tree1)
```

```
##
```

```
## Data: pred_tree1 in 1140 controls (validation_data$churn 0) < 193 cases (validation_data$churn 1).
```

```
## Area under the curve: 0.6378
```

```
plot.roc(validation_data$churn,pred_tree1,col = "red", lwd = 3)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

