Exploratory Data Analysis

Importing the libraries

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

library(caTools)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

Reading the dataset

data<-read.csv("Placement\_Data\_Full\_Class.csv")  
head(data)

## sl\_no gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex  
## 1 1 M 67.00 Others 91.00 Others Commerce 58.00 Sci&Tech No  
## 2 2 M 79.33 Central 78.33 Others Science 77.48 Sci&Tech Yes  
## 3 3 M 65.00 Central 68.00 Central Arts 64.00 Comm&Mgmt No  
## 4 4 M 56.00 Central 52.00 Central Science 52.00 Sci&Tech No  
## 5 5 M 85.80 Central 73.60 Central Commerce 73.30 Comm&Mgmt No  
## 6 6 M 55.00 Others 49.80 Others Science 67.25 Sci&Tech Yes  
## etest\_p specialisation mba\_p status salary  
## 1 55.0 Mkt&HR 58.80 Placed 270000  
## 2 86.5 Mkt&Fin 66.28 Placed 200000  
## 3 75.0 Mkt&Fin 57.80 Placed 250000  
## 4 66.0 Mkt&HR 59.43 Not Placed NA  
## 5 96.8 Mkt&Fin 55.50 Placed 425000  
## 6 55.0 Mkt&Fin 51.58 Not Placed NA

Data Cleaning and Data Pre-processing 1.Dropping the first column

data<-select(data,-1)  
head(data)

## gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex etest\_p  
## 1 M 67.00 Others 91.00 Others Commerce 58.00 Sci&Tech No 55.0  
## 2 M 79.33 Central 78.33 Others Science 77.48 Sci&Tech Yes 86.5  
## 3 M 65.00 Central 68.00 Central Arts 64.00 Comm&Mgmt No 75.0  
## 4 M 56.00 Central 52.00 Central Science 52.00 Sci&Tech No 66.0  
## 5 M 85.80 Central 73.60 Central Commerce 73.30 Comm&Mgmt No 96.8  
## 6 M 55.00 Others 49.80 Others Science 67.25 Sci&Tech Yes 55.0  
## specialisation mba\_p status salary  
## 1 Mkt&HR 58.80 Placed 270000  
## 2 Mkt&Fin 66.28 Placed 200000  
## 3 Mkt&Fin 57.80 Placed 250000  
## 4 Mkt&HR 59.43 Not Placed NA  
## 5 Mkt&Fin 55.50 Placed 425000  
## 6 Mkt&Fin 51.58 Not Placed NA

2.Removing Null Values

data$salary[is.na(data$salary)]<-0  
head(data)

## gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex etest\_p  
## 1 M 67.00 Others 91.00 Others Commerce 58.00 Sci&Tech No 55.0  
## 2 M 79.33 Central 78.33 Others Science 77.48 Sci&Tech Yes 86.5  
## 3 M 65.00 Central 68.00 Central Arts 64.00 Comm&Mgmt No 75.0  
## 4 M 56.00 Central 52.00 Central Science 52.00 Sci&Tech No 66.0  
## 5 M 85.80 Central 73.60 Central Commerce 73.30 Comm&Mgmt No 96.8  
## 6 M 55.00 Others 49.80 Others Science 67.25 Sci&Tech Yes 55.0  
## specialisation mba\_p status salary  
## 1 Mkt&HR 58.80 Placed 270000  
## 2 Mkt&Fin 66.28 Placed 200000  
## 3 Mkt&Fin 57.80 Placed 250000  
## 4 Mkt&HR 59.43 Not Placed 0  
## 5 Mkt&Fin 55.50 Placed 425000  
## 6 Mkt&Fin 51.58 Not Placed 0

3.Normalization

normalize<-function(attribute){ return ( (attribute - min(attribute))/(max(attribute)-min(attribute))) }  
data$ssc\_p<-normalize(data$ssc\_p)  
data$hsc\_p<-normalize(data$hsc\_p)  
data$degree\_p<-normalize(data$degree\_p)  
data$etest\_p<-normalize(data$etest\_p)  
data$mba\_p<-normalize(data$mba\_p)  
data$salary<-normalize(data$salary)  
head(data)

## gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t  
## 1 M 0.5382395 Others 0.8896211 Others Commerce 0.19512195 Sci&Tech  
## 2 M 0.7924139 Central 0.6808896 Others Science 0.67024390 Sci&Tech  
## 3 M 0.4970109 Central 0.5107084 Central Arts 0.34146341 Comm&Mgmt  
## 4 M 0.3114822 Central 0.2471170 Central Science 0.04878049 Sci&Tech  
## 5 M 0.9257885 Central 0.6029654 Central Commerce 0.56829268 Comm&Mgmt  
## 6 M 0.2908679 Others 0.2108731 Others Science 0.42073171 Sci&Tech  
## workex etest\_p specialisation mba\_p status salary  
## 1 No 0.1041667 Mkt&HR 0.28448276 Placed 0.2872340  
## 2 Yes 0.7604167 Mkt&Fin 0.56484258 Placed 0.2127660  
## 3 No 0.5208333 Mkt&Fin 0.24700150 Placed 0.2659574  
## 4 No 0.3333333 Mkt&HR 0.30809595 Not Placed 0.0000000  
## 5 No 0.9750000 Mkt&Fin 0.16079460 Placed 0.4521277  
## 6 Yes 0.1041667 Mkt&Fin 0.01386807 Not Placed 0.0000000

Data preview

str(data)

## 'data.frame': 215 obs. of 14 variables:  
## $ gender : chr "M" "M" "M" "M" ...  
## $ ssc\_p : num 0.538 0.792 0.497 0.311 0.926 ...  
## $ ssc\_b : chr "Others" "Central" "Central" "Central" ...  
## $ hsc\_p : num 0.89 0.681 0.511 0.247 0.603 ...  
## $ hsc\_b : chr "Others" "Others" "Central" "Central" ...  
## $ hsc\_s : chr "Commerce" "Science" "Arts" "Science" ...  
## $ degree\_p : num 0.1951 0.6702 0.3415 0.0488 0.5683 ...  
## $ degree\_t : chr "Sci&Tech" "Sci&Tech" "Comm&Mgmt" "Sci&Tech" ...  
## $ workex : chr "No" "Yes" "No" "No" ...  
## $ etest\_p : num 0.104 0.76 0.521 0.333 0.975 ...  
## $ specialisation: chr "Mkt&HR" "Mkt&Fin" "Mkt&Fin" "Mkt&HR" ...  
## $ mba\_p : num 0.284 0.565 0.247 0.308 0.161 ...  
## $ status : chr "Placed" "Placed" "Placed" "Not Placed" ...  
## $ salary : num 0.287 0.213 0.266 0 0.452 ...

Handling categorical variables

data$status=as.numeric(as.factor(data$status))-1  
data$gender=as.numeric(as.factor(data$gender))-1  
data$specialisation=as.numeric(as.factor(data$specialisation))-1  
data$ssc\_b=as.numeric(as.factor(data$ssc\_b))-1  
data$hsc\_b=as.numeric(as.factor(data$hsc\_b))-1  
data$hsc\_s=as.numeric(as.factor(data$hsc\_s))-1  
data$degree\_t=as.numeric(as.factor(data$degree\_t))-1  
data$workex=as.numeric(as.factor(data$workex))-1  
data$status=as.factor(data$status)  
data$workex=as.factor(data$workex)  
data$specialisation=as.factor(data$specialisation)  
data$gender=as.factor(data$gender)  
data$ssc\_b=as.factor(data$ssc\_b)  
data$hsc\_b=as.factor(data$hsc\_b)  
data$hsc\_s=as.factor(data$hsc\_s)  
data$degree\_t=as.factor(data$degree\_t)  
str(data)

## 'data.frame': 215 obs. of 14 variables:  
## $ gender : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 1 2 2 2 ...  
## $ ssc\_p : num 0.538 0.792 0.497 0.311 0.926 ...  
## $ ssc\_b : Factor w/ 2 levels "0","1": 2 1 1 1 1 2 2 1 1 1 ...  
## $ hsc\_p : num 0.89 0.681 0.511 0.247 0.603 ...  
## $ hsc\_b : Factor w/ 2 levels "0","1": 2 2 1 1 1 2 2 1 1 1 ...  
## $ hsc\_s : Factor w/ 3 levels "0","1","2": 2 3 1 3 2 3 2 3 2 2 ...  
## $ degree\_p : num 0.1951 0.6702 0.3415 0.0488 0.5683 ...  
## $ degree\_t : Factor w/ 3 levels "0","1","2": 3 3 1 3 1 3 1 3 1 1 ...  
## $ workex : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 1 2 1 1 ...  
## $ etest\_p : num 0.104 0.76 0.521 0.333 0.975 ...  
## $ specialisation: Factor w/ 2 levels "0","1": 2 1 1 2 1 1 1 1 1 1 ...  
## $ mba\_p : num 0.284 0.565 0.247 0.308 0.161 ...  
## $ status : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 1 2 2 1 ...  
## $ salary : num 0.287 0.213 0.266 0 0.452 ...

Swapping the last two columns ( predictor variable is moved to the last column )

data=data[,c(1,2,3,4,5,6,7,8,9,10,11,12,14,13)]  
head(data)

## gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex  
## 1 1 0.5382395 1 0.8896211 1 1 0.19512195 2 0  
## 2 1 0.7924139 0 0.6808896 1 2 0.67024390 2 1  
## 3 1 0.4970109 0 0.5107084 0 0 0.34146341 0 0  
## 4 1 0.3114822 0 0.2471170 0 2 0.04878049 2 0  
## 5 1 0.9257885 0 0.6029654 0 1 0.56829268 0 0  
## 6 1 0.2908679 1 0.2108731 1 2 0.42073171 2 1  
## etest\_p specialisation mba\_p salary status  
## 1 0.1041667 1 0.28448276 0.2872340 1  
## 2 0.7604167 0 0.56484258 0.2127660 1  
## 3 0.5208333 0 0.24700150 0.2659574 1  
## 4 0.3333333 1 0.30809595 0.0000000 0  
## 5 0.9750000 0 0.16079460 0.4521277 1  
## 6 0.1041667 0 0.01386807 0.0000000 0

Splitting the dataset into training and test data ( In the ratio 67:33 )

data\_set\_size=floor(nrow(data)\*0.67)  
index<-sample(1:nrow(data),size=data\_set\_size)  
train<-data[index,]  
test<-data[-index,]

test

## gender ssc\_p ssc\_b hsc\_p hsc\_b hsc\_s degree\_p degree\_t workex  
## 1 1 0.5382395 1 0.88962109 1 1 0.195121951 2 0  
## 3 1 0.4970109 0 0.51070840 0 0 0.341463415 0 0  
## 7 0 0.1053391 1 0.20098847 1 1 0.707317073 0 0  
## 9 1 0.6619254 0 0.69192751 0 1 0.536585366 0 0  
## 10 1 0.3527108 0 0.54365733 0 1 0.268292683 0 0  
## 13 0 0.1259534 0 0.29654036 1 2 0.365853659 0 0  
## 14 0 0.7443826 0 0.82372323 0 1 0.219512195 0 0  
## 19 0 0.4557823 0 0.47775947 0 1 0.341463415 0 0  
## 24 0 0.7526283 1 0.37891269 1 2 0.359512195 2 1  
## 25 1 0.7340754 1 1.00000000 1 2 0.703902439 2 0  
## 26 0 0.2409812 1 0.28995058 0 1 0.004878049 0 1  
## 31 0 0.4763966 0 0.60131796 0 1 0.560975610 0 0  
## 34 0 0.9505257 1 0.46128501 1 2 0.756097561 0 1  
## 36 0 0.5794682 0 0.67545305 0 1 0.536585366 0 0  
## 40 1 0.8268398 1 0.51070840 1 2 0.341463415 2 0  
## 42 0 0.6825397 1 0.43097199 1 1 0.365853659 0 1  
## 45 0 0.7443826 1 0.59308072 1 1 0.756097561 0 1  
## 52 1 0.2784993 0 0.39736409 0 1 0.151219512 0 0  
## 56 1 0.4021851 0 0.48764415 1 2 0.365853659 0 0  
## 61 1 0.6825397 0 0.54365733 0 2 0.536585366 0 1  
## 63 0 0.9402185 1 0.44810544 1 2 0.424390244 2 0  
## 68 1 0.8251907 1 0.68369028 1 1 0.414634146 0 0  
## 71 1 0.8474541 1 0.39538715 1 2 0.292682927 2 0  
## 72 1 0.7031540 1 0.54843493 1 1 0.512195122 0 0  
## 74 1 0.4887652 0 0.77149918 1 1 0.529756098 0 0  
## 75 1 0.3238508 0 0.45799012 0 1 0.492682927 0 0  
## 76 0 0.3733251 0 0.41186161 1 1 0.670731707 0 0  
## 77 0 0.5279324 1 0.55024712 0 0 0.534878049 0 0  
## 78 1 0.4763966 1 0.70840198 1 2 0.365853659 2 1  
## 79 1 0.8886827 1 0.88797364 1 2 0.353658537 2 0  
## 81 0 0.5794682 1 0.41186161 1 1 0.463414634 0 1  
## 82 1 0.8412698 1 0.42833608 1 2 0.414634146 0 1  
## 83 1 0.4557823 0 0.49423394 0 1 0.585365854 0 0  
## 88 1 0.3856937 0 0.23064250 0 2 0.243902439 1 0  
## 92 1 0.2290249 0 0.32948929 0 1 0.019512195 0 0  
## 93 0 0.3986807 0 0.52718287 0 2 0.390243902 0 0  
## 94 1 0.2290249 0 0.41186161 0 1 0.097560976 0 0  
## 95 1 0.3527108 0 0.41186161 0 1 0.341463415 0 0  
## 98 0 0.6103896 0 0.42009885 1 1 0.268292683 0 0  
## 99 0 0.5794682 0 0.59308072 0 1 0.365853659 0 0  
## 100 1 0.2702536 0 0.74135091 1 1 0.317073171 2 0  
## 102 1 0.4557823 0 0.57660626 0 1 0.439024390 0 0  
## 103 0 0.7443826 1 0.39538715 1 1 0.439024390 0 1  
## 107 1 0.4162028 1 0.21416804 1 2 0.097560976 2 0  
## 116 0 0.6619254 1 0.42833608 1 2 0.390243902 0 0  
## 118 1 0.7443826 1 0.62602965 1 2 0.560975610 2 0  
## 119 1 0.7237683 0 0.70840198 0 2 0.682926829 2 1  
## 121 1 0.3527108 1 0.04942339 1 2 0.219512195 0 0  
## 124 1 0.6825397 1 0.36243822 1 1 0.560975610 0 1  
## 129 1 0.8144712 0 0.59967051 0 2 0.676097561 2 1  
## 131 1 0.4351680 0 0.46128501 1 1 0.243902439 0 0  
## 134 1 0.6619254 0 0.44481054 1 1 0.658536585 0 1  
## 138 1 0.5382395 1 0.42833608 0 1 0.536585366 0 0  
## 140 1 0.7443826 0 0.54365733 0 1 0.219512195 0 1  
## 142 1 0.5176252 0 0.44481054 0 2 0.243902439 0 0  
## 145 1 0.2290249 1 0.21416804 1 0 0.268292683 0 0  
## 147 1 0.4351680 0 0.42833608 1 2 0.390243902 0 0  
## 152 1 0.4970109 0 0.46128501 0 1 0.609756098 0 0  
## 154 1 0.1671820 1 0.36243822 1 2 0.365853659 2 1  
## 156 1 0.2201608 1 0.62042834 1 1 0.241463415 0 1  
## 160 1 0.2290249 0 0.19769357 1 1 0.195121951 0 0  
## 166 0 0.4619666 0 0.68088962 1 1 0.585365854 0 0  
## 168 1 0.5567924 1 0.41186161 1 2 0.414634146 2 1  
## 177 0 0.3733251 0 0.37891269 1 1 0.146341463 0 0  
## 179 1 0.5588538 1 0.31301483 1 2 0.439024390 2 0  
## 191 0 0.4763966 1 0.54695222 0 1 0.268292683 0 0  
## 197 1 0.6413111 1 0.42833608 1 2 0.670731707 2 1  
## 199 0 0.5382395 0 0.54365733 0 1 0.365853659 1 0  
## 201 1 0.5794682 1 0.37891269 1 1 0.365853659 0 0  
## 206 1 0.4145537 1 0.41186161 1 1 0.365853659 0 0  
## 215 1 0.4351680 0 0.34596376 1 2 0.073170732 0 0  
## etest\_p specialisation mba\_p salary status  
## 1 0.10416667 1 0.28448276 0.2872340 1  
## 3 0.52083333 0 0.24700150 0.2659574 1  
## 7 0.50583333 0 0.07796102 0.0000000 0  
## 9 0.86125000 0 0.37781109 0.2457447 1  
## 10 0.08333333 0 0.03748126 0.0000000 0  
## 13 0.25000000 1 0.51836582 0.0000000 0  
## 14 0.37500000 0 0.65292354 0.2319149 1  
## 19 0.37500000 1 0.48238381 0.0000000 0  
## 24 0.87500000 0 0.46514243 0.3191489 1  
## 25 0.98750000 0 0.85457271 0.3829787 1  
## 26 0.54166667 0 0.52923538 0.0000000 0  
## 31 0.04166667 1 0.20577211 0.2659574 1  
## 34 0.79166667 0 0.80847076 0.2765957 1  
## 36 0.43750000 1 0.43215892 0.3191489 1  
## 40 0.89583333 0 0.42541229 0.4372340 1  
## 42 0.31250000 1 0.69527736 0.0000000 0  
## 45 0.81250000 0 0.69302849 0.2127660 1  
## 52 0.35416667 1 0.42878561 0.0000000 0  
## 56 0.43750000 1 0.05622189 0.2340426 1  
## 61 0.20833333 0 0.22601199 0.2765957 1  
## 63 0.18750000 0 0.31784108 0.2553191 1  
## 68 0.38979167 0 0.36656672 0.2925532 1  
## 71 0.81250000 0 0.53373313 0.3829787 1  
## 72 0.93750000 0 0.58958021 0.2553191 1  
## 74 0.75000000 0 0.32008996 0.2319149 1  
## 75 0.71395833 0 0.59932534 0.3574468 1  
## 76 0.50000000 1 0.59182909 0.0000000 0  
## 77 0.22916667 0 0.48950525 0.2446809 1  
## 78 0.39583333 0 0.24137931 0.5319149 1  
## 79 0.75083333 0 0.30772114 0.2872340 1  
## 81 0.35416667 1 0.41754123 0.2553191 1  
## 82 0.75000000 0 0.71176912 0.3191489 1  
## 83 0.66666667 0 0.34595202 0.0000000 0  
## 88 0.52083333 1 0.29497751 0.0000000 0  
## 92 0.35416667 1 0.43403298 0.0000000 0  
## 93 0.45833333 0 0.30959520 0.2446809 1  
## 94 0.45833333 1 0.15742129 0.0000000 0  
## 95 0.08083333 0 0.14092954 0.2765957 1  
## 98 0.91479167 0 0.66791604 0.0000000 0  
## 99 0.41666667 0 0.22863568 0.2340426 1  
## 100 0.00000000 0 0.30959520 0.0000000 0  
## 102 0.58333333 1 0.34595202 0.4042553 1  
## 103 0.15625000 0 0.37856072 0.3191489 1  
## 107 0.43750000 0 0.54272864 0.0000000 0  
## 116 0.81250000 0 0.34820090 0.2297872 1  
## 118 0.62500000 0 0.59370315 0.2553191 1  
## 119 0.97916667 1 0.72226387 0.2936170 1  
## 121 0.47916667 1 0.28485757 0.0000000 0  
## 124 0.20833333 1 0.20577211 0.2553191 1  
## 129 0.65000000 1 0.93890555 0.4255319 1  
## 131 0.70833333 0 0.48500750 0.0000000 0  
## 134 0.31250000 1 0.36619190 0.2659574 1  
## 138 0.12500000 1 0.34482759 0.2393617 1  
## 140 0.16666667 0 0.12068966 0.2340426 1  
## 142 0.20833333 1 0.40067466 0.0000000 0  
## 145 0.20833333 0 0.27398801 0.0000000 0  
## 147 0.72916667 1 0.14730135 0.2478723 1  
## 152 0.68750000 0 0.28710645 0.2872340 1  
## 154 0.75000000 0 0.42241379 0.3617021 1  
## 156 0.12812500 1 0.55397301 0.0000000 0  
## 160 0.25000000 1 0.35157421 0.0000000 0  
## 166 0.62500000 0 0.87518741 0.0000000 0  
## 168 0.16875000 0 0.91829085 0.0000000 0  
## 177 0.10416667 1 0.25074963 0.2340426 1  
## 179 0.47916667 1 0.63193403 0.3723404 1  
## 191 0.00000000 0 0.42316342 0.0000000 0  
## 197 0.58333333 0 0.12256372 0.2659574 1  
## 199 0.79166667 1 0.77773613 0.0000000 0  
## 201 0.78229167 0 0.05997001 0.3191489 1  
## 206 0.25000000 0 0.20989505 0.2659574 1  
## 215 0.81250000 1 0.33770615 0.0000000 0

Building the random forest model :

1.Initially we set the ntree=1 , this implies that a single decision tree is used to predict the status. Let us check the accuracy of this model and then proceed with the construction of multiple decision trees.

set.seed(80)  
rfd <-randomForest(status~.,data=train, ntree=1,mtry=3,importance=TRUE)  
rfd

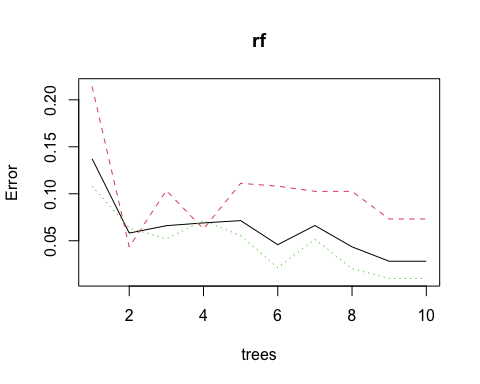
##   
## Call:  
## randomForest(formula = status ~ ., data = train, ntree = 1, mtry = 3, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 1  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 4%  
## Confusion matrix:  
## 0 1 class.error  
## 0 13 0 0.00000000  
## 1 2 35 0.05405405

1. The error rate of a single decision tree is displayed above. Let us now increase the ntree value and find out the accuracy and the error rate of the random forest model.

rf <-randomForest(status~.,data=train, ntree= 10,mtry=3,importance=TRUE)  
rf

##   
## Call:  
## randomForest(formula = status ~ ., data = train, ntree = 10, mtry = 3, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 10  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 2.82%  
## Confusion matrix:  
## 0 1 class.error  
## 0 38 3 0.07317073  
## 1 1 100 0.00990099

plot(rf)

 3. We can see clearly that using a random forest model over a single decision tree has decreased the error rate and thereby increased the accuracy.

Evaluating the model by running it on the test dataset :

res<-predict(rf,newdata=test,type="class")  
confusionMatrix(table(res,test$status))

## Confusion Matrix and Statistics  
##   
##   
## res 0 1  
## 0 25 0  
## 1 1 45  
##   
## Accuracy : 0.9859   
## 95% CI : (0.924, 0.9996)  
## No Information Rate : 0.6338   
## P-Value [Acc > NIR] : 3.649e-13   
##   
## Kappa : 0.9694   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.9615   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9783   
## Prevalence : 0.3662   
## Detection Rate : 0.3521   
## Detection Prevalence : 0.3521   
## Balanced Accuracy : 0.9808   
##   
## 'Positive' Class : 0   
##