Predicting Insurance Fraud

claims<-read.csv("D:\\CSUEB\_MSBA\\Capstone\\claims.csv")

dim(claims)

## [1] 15420 33

summary(claims)

## Month WeekOfMonth DayOfWeek Make   
## Jan :1411 Min. :1.000 Friday :2445 Pontiac :3837   
## May :1367 1st Qu.:2.000 Monday :2616 Toyota :3121   
## Mar :1360 Median :3.000 Saturday :1982 Honda :2801   
## Jun :1321 Mean :2.789 Sunday :1745 Mazda :2354   
## Oct :1305 3rd Qu.:4.000 Thursday :2173 Chevrolet:1681   
## Dec :1285 Max. :5.000 Tuesday :2300 Accura : 472   
## (Other):7371 Wednesday:2159 (Other) :1154   
## AccidentArea DayOfWeekClaimed MonthClaimed WeekOfMonthClaimed  
## Rural: 1598 Monday :3757 Jan :1446 Min. :1.000   
## Urban:13822 Tuesday :3375 May :1411 1st Qu.:2.000   
## Wednesday:2951 Mar :1348 Median :3.000   
## Thursday :2660 Oct :1339 Mean :2.694   
## Friday :2497 Jun :1293 3rd Qu.:4.000   
## Saturday : 127 Feb :1287 Max. :5.000   
## (Other) : 53 (Other):7296   
## Sex MaritalStatus Age Fault   
## Female: 2420 Divorced: 76 Min. : 0.00 Policy Holder:11230   
## Male :13000 Married :10625 1st Qu.:31.00 Third Party : 4190   
## Single : 4684 Median :38.00   
## Widow : 35 Mean :39.86   
## 3rd Qu.:48.00   
## Max. :80.00   
##   
## PolicyType VehicleCategory VehiclePrice   
## Sedan - Collision :5584 Sedan :9671 20000 to 29000 :8079   
## Sedan - Liability :4987 Sport :5358 30000 to 39000 :3533   
## Sedan - All Perils :4087 Utility: 391 40000 to 59000 : 461   
## Sport - Collision : 348 60000 to 69000 : 87   
## Utility - All Perils: 340 less than 20000:1096   
## Utility - Collision : 30 more than 69000:2164   
## (Other) : 44   
## FraudFound\_P PolicyNumber RepNumber Deductible   
## Min. :0.00000 Min. : 1 Min. : 1.000 Min. :300.0   
## 1st Qu.:0.00000 1st Qu.: 3856 1st Qu.: 5.000 1st Qu.:400.0   
## Median :0.00000 Median : 7710 Median : 8.000 Median :400.0   
## Mean :0.05986 Mean : 7710 Mean : 8.483 Mean :407.7   
## 3rd Qu.:0.00000 3rd Qu.:11565 3rd Qu.:12.000 3rd Qu.:400.0   
## Max. :1.00000 Max. :15420 Max. :16.000 Max. :700.0   
##   
## DriverRating Days\_Policy\_Accident Days\_Policy\_Claim  
## Min. :1.000 1 to 7 : 14 15 to 30 : 56   
## 1st Qu.:1.000 15 to 30 : 49 8 to 15 : 21   
## Median :2.000 8 to 15 : 55 more than 30:15342   
## Mean :2.488 more than 30:15247 none : 1   
## 3rd Qu.:3.000 none : 55   
## Max. :4.000   
##   
## PastNumberOfClaims AgeOfVehicle AgeOfPolicyHolder  
## 1 :3573 7 years :5807 31 to 35:5593   
## 2 to 4 :5485 more than 7:3981 36 to 40:4043   
## more than 4:2010 6 years :3448 41 to 50:2828   
## none :4352 5 years :1357 51 to 65:1392   
## new : 373 26 to 30: 613   
## 4 years : 229 over 65 : 508   
## (Other) : 225 (Other) : 443   
## PoliceReportFiled WitnessPresent AgentType NumberOfSuppliments  
## No :14992 No :15333 External:15179 1 to 2 :2489   
## Yes: 428 Yes: 87 Internal: 241 3 to 5 :2017   
## more than 5:3867   
## none :7047   
##   
##   
##   
## AddressChange\_Claim NumberOfCars Year   
## 1 year : 170 1 vehicle :14316 Min. :1994   
## 2 to 3 years : 291 2 vehicles : 709 1st Qu.:1994   
## 4 to 8 years : 631 3 to 4 : 372 Median :1995   
## no change :14324 5 to 8 : 21 Mean :1995   
## under 6 months: 4 more than 8: 2 3rd Qu.:1996   
## Max. :1996   
##   
## BasePolicy   
## All Perils:4449   
## Collision :5962   
## Liability :5009   
##   
##   
##   
##

#convert numeric to factors  
claims$Year <- as.factor(claims$Year)  
claims$RepNumber <- as.factor(claims$RepNumber)  
claims$WeekOfMonth <- as.factor(claims$WeekOfMonth)  
claims$WeekOfMonthClaimed <- as.factor(claims$WeekOfMonthClaimed)  
claims$FraudFound\_P <- as.factor(claims$FraudFound\_P)  
  
summary(claims$AgeOfPolicyHolder[which(claims$Age == 0)])

## 16 to 17 18 to 20 21 to 25 26 to 30 31 to 35 36 to 40 41 to 50 51 to 65   
## 320 0 0 0 0 0 0 0   
## over 65   
## 0

#Since Age and AgeOfPolicyHolder variables both are the same and as Age variable has missing data, we will drop it. Moreover, variable AgeOfPolicyHolder is already divided into bins  
claims <-claims[-11]

## check if there are any missing values in the dataset

sum(is.na(claims))

## [1] 0

## no missing values found

#Policy number will also cause over-fitting, hence remove it  
  
claims<-claims[,-c(16)]

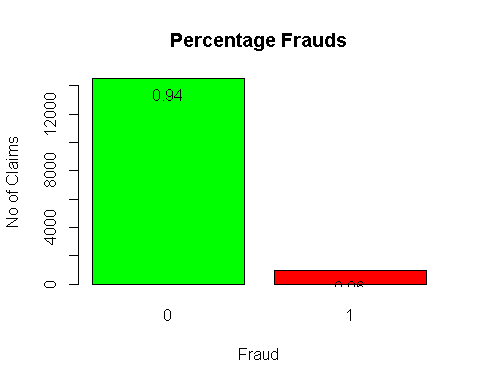
claims$FraudFound\_P <-as.factor(claims$FraudFound\_P)#encode as factor  
table(claims$FraudFound\_P)#see distribution of Response variable- Fraud\_Found\_P

##   
## 0 1   
## 14497 923

contrasts(claims$FraudFound\_P) #see how factor is encoded

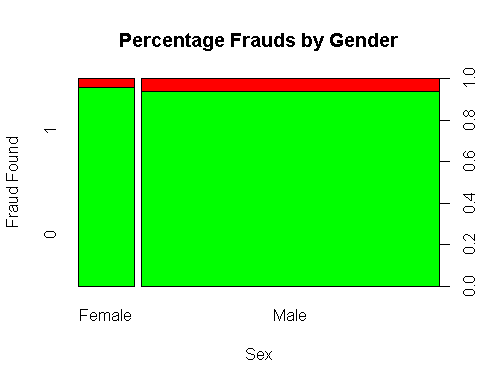
## 1  
## 0 0  
## 1 1

count <- table(claims$FraudFound\_P)   
prop <- round(count/length(claims$FraudFound\_P),2)  
b <- plot(claims$FraudFound\_P, xlab='Fraud', ylab='No of Claims', main='Percentage Frauds', legend = rownames(count),beside=TRUE, col=c("green","red"))   
text(b, count, prop, pos=1)

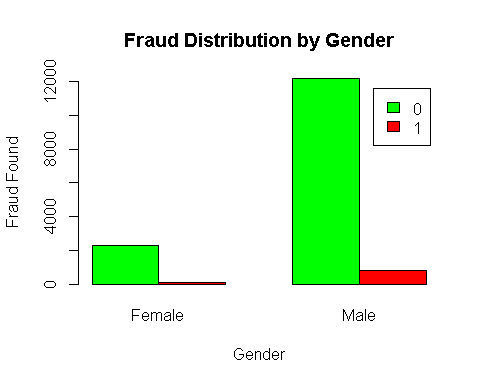
 ##923 frauds were found out of 15420 samples ## 94% claims were not fraudulent ##Data is highly imbalanced

# Exploratory ANalysis

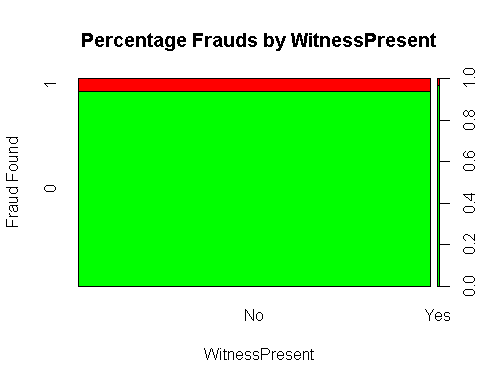
b <- plot(claims$Sex,claims$FraudFound\_P , ylab='Fraud Found', xlab='Sex', main='Percentage Frauds by Gender', col=c("green","red"))



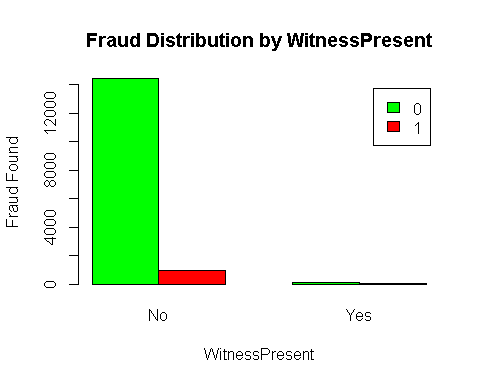
counts <- table(claims$FraudFound\_P,claims$Sex)  
barplot(counts, main="Fraud Distribution by Gender",  
 ylab="Fraud Found", xlab= "Gender",  
 legend = rownames(counts),beside=TRUE, col=c("green","red"))  
text(b, counts, prop, pos=1)



b <- plot(claims$WitnessPresent,claims$FraudFound\_P , ylab='Fraud Found', xlab='WitnessPresent', main='Percentage Frauds by WitnessPresent', col=c("green","red"))



counts <- table(claims$FraudFound\_P,claims$WitnessPresent)  
barplot(counts, main="Fraud Distribution by WitnessPresent",  
 ylab="Fraud Found", xlab= "WitnessPresent",  
 legend = rownames(counts),beside=TRUE, col=c("green","red"))  
text(b, counts, prop, pos=1)



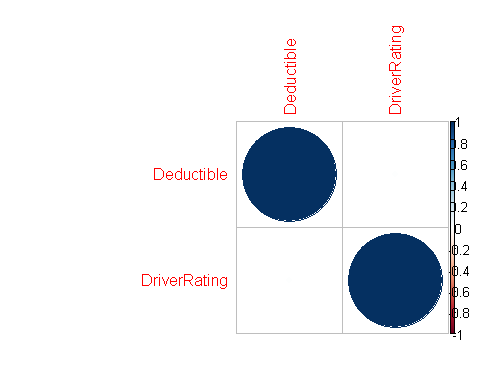
#extract numeric variables  
claims\_num <-claims[,sapply(claims,is.numeric)]  
#claims\_num <-claims[,colnames]

#see correlation between the numeric predictors  
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.5.3

## corrplot 0.84 loaded

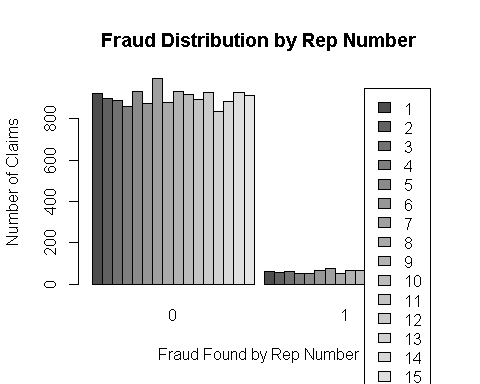
correlations <- cor(claims\_num)  
corrplot(correlations, method="circle")



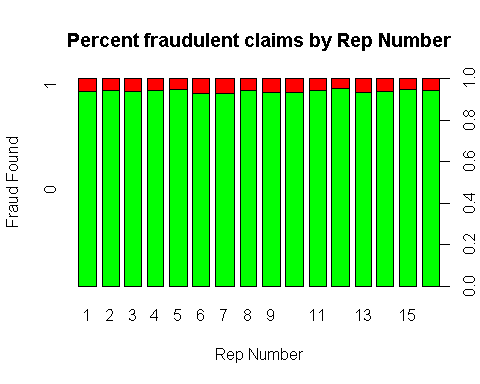
# We do not get any additional insights from the two numeric variables

#pairs(claims\_num, claims$FraudFound\_P)

# Stacked Bar Plot with Colors and Legend  
counts <- table(claims$RepNumber, claims$FraudFound\_P)  
barplot(counts, main="Fraud Distribution by Rep Number",  
 xlab="Fraud Found by Rep Number", ylab= "Number of Claims",  
 legend = rownames(counts),beside=TRUE)



plot(claims$RepNumber, claims$FraudFound\_P, main= "Percent fraudulent claims by Rep Number", ylab="Fraud Found", xlab= "Rep Number",col=c("green","red"))



set.seed(123)  
#split into training (65%) and test(35%) data by random partitioning  
x <- claims[, -15]  
y <- claims$FraudFound\_P  
n <- nrow(x)  
  
train <- sample(1:n, floor(0.65 \* n))  
  
y\_test <- y[-train]  
y\_train <- y[train]  
x\_test <- x[-train, ]  
x\_train <- x[train, ]

## Multiple logistic regression model

glm\_full <- glm(y\_train ~ ., data=x\_train, family=binomial,maxit = 100)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(glm\_full)

##   
## Call:  
## glm(formula = y\_train ~ ., family = binomial, data = x\_train,   
## maxit = 100)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5206 -0.4248 -0.1598 -0.1049 3.3641   
##   
## Coefficients: (5 not defined because of singularities)  
## Estimate Std. Error z value  
## (Intercept) -3.080e+12 1.482e+13 -2.080e-01  
## MonthAug -2.433e-01 3.853e-01 -6.310e-01  
## MonthDec -7.168e-02 3.880e-01 -1.850e-01  
## MonthFeb -3.572e-02 3.398e-01 -1.050e-01  
## MonthJan -9.775e-02 3.599e-01 -2.720e-01  
## MonthJul -3.157e-01 3.603e-01 -8.760e-01  
## MonthJun 1.257e-01 3.306e-01 3.800e-01  
## MonthMar 4.804e-01 2.619e-01 1.834e+00  
## MonthMay -1.175e-01 2.649e-01 -4.440e-01  
## MonthNov 1.121e-01 4.162e-01 2.690e-01  
## MonthOct 4.237e-02 4.119e-01 1.030e-01  
## MonthSep -6.043e-02 4.011e-01 -1.510e-01  
## WeekOfMonth2 1.486e-01 1.384e-01 1.074e+00  
## WeekOfMonth3 -1.689e-02 1.480e-01 -1.140e-01  
## WeekOfMonth4 -8.584e-02 1.481e-01 -5.800e-01  
## WeekOfMonth5 -1.598e-01 1.794e-01 -8.910e-01  
## DayOfWeekMonday -9.348e-02 1.540e-01 -6.070e-01  
## DayOfWeekSaturday -9.363e-02 1.605e-01 -5.830e-01  
## DayOfWeekSunday 5.075e-02 1.650e-01 3.080e-01  
## DayOfWeekThursday -1.333e-01 1.618e-01 -8.240e-01  
## DayOfWeekTuesday -1.125e-01 1.593e-01 -7.060e-01  
## DayOfWeekWednesday -1.212e-01 1.650e-01 -7.340e-01  
## MakeBMW -4.504e+15 2.122e+07 -2.122e+08  
## MakeChevrolet -4.966e-01 2.372e-01 -2.093e+00  
## MakeDodge -8.816e-01 7.747e-01 -1.138e+00  
## MakeFerrari -4.504e+15 6.711e+07 -6.711e+07  
## MakeFord -2.001e-01 3.034e-01 -6.590e-01  
## MakeHonda -4.039e-01 2.353e-01 -1.716e+00  
## MakeJaguar -4.504e+15 3.356e+07 -1.342e+08  
## MakeLexus -4.504e+15 6.711e+07 -6.711e+07  
## MakeMazda -4.143e-01 2.331e-01 -1.777e+00  
## MakeMecedes -4.504e+15 4.745e+07 -9.490e+07  
## MakeMercury -6.512e-01 5.703e-01 -1.142e+00  
## MakeNisson -7.403e-01 1.077e+00 -6.870e-01  
## MakePontiac -4.775e-01 2.214e-01 -2.157e+00  
## MakePorche -4.504e+15 3.355e+07 -1.342e+08  
## MakeSaab -4.857e-02 4.435e-01 -1.100e-01  
## MakeSaturn -7.688e-01 6.571e-01 -1.170e+00  
## MakeToyota -3.434e-01 2.253e-01 -1.524e+00  
## MakeVW -1.245e+00 4.739e-01 -2.628e+00  
## AccidentAreaUrban -1.655e-01 1.315e-01 -1.258e+00  
## DayOfWeekClaimedFriday 3.080e+12 1.482e+13 2.080e-01  
## DayOfWeekClaimedMonday 3.080e+12 1.482e+13 2.080e-01  
## DayOfWeekClaimedSaturday 3.080e+12 1.482e+13 2.080e-01  
## DayOfWeekClaimedSunday 3.080e+12 1.482e+13 2.080e-01  
## DayOfWeekClaimedThursday 3.080e+12 1.482e+13 2.080e-01  
## DayOfWeekClaimedTuesday 3.080e+12 1.482e+13 2.080e-01  
## DayOfWeekClaimedWednesday 3.080e+12 1.482e+13 2.080e-01  
## MonthClaimedApr -2.848e-01 3.997e-01 -7.130e-01  
## MonthClaimedAug 5.192e-01 2.977e-01 1.744e+00  
## MonthClaimedDec -5.457e-01 4.167e-01 -1.309e+00  
## MonthClaimedFeb -3.136e-01 4.198e-01 -7.470e-01  
## MonthClaimedJan -1.017e-01 4.057e-01 -2.510e-01  
## MonthClaimedJul -7.241e-02 3.458e-01 -2.090e-01  
## MonthClaimedJun -3.099e-02 3.728e-01 -8.300e-02  
## MonthClaimedMar -3.460e-01 4.054e-01 -8.540e-01  
## MonthClaimedMay 4.327e-02 3.915e-01 1.110e-01  
## MonthClaimedNov -9.400e-01 4.105e-01 -2.290e+00  
## MonthClaimedOct -1.107e-01 3.139e-01 -3.520e-01  
## MonthClaimedSep NA NA NA  
## WeekOfMonthClaimed2 -2.902e-01 1.398e-01 -2.076e+00  
## WeekOfMonthClaimed3 -1.305e-01 1.466e-01 -8.900e-01  
## WeekOfMonthClaimed4 3.495e-02 1.422e-01 2.460e-01  
## WeekOfMonthClaimed5 -2.355e-01 1.995e-01 -1.180e+00  
## SexMale 3.012e-01 1.391e-01 2.165e+00  
## MaritalStatusMarried 5.696e-02 6.280e-01 9.100e-02  
## MaritalStatusSingle -1.467e-02 6.341e-01 -2.300e-02  
## MaritalStatusWidow 9.111e-01 1.041e+00 8.750e-01  
## FaultThird Party -2.879e+00 2.203e-01 -1.306e+01  
## PolicyTypeSedan - Collision -5.457e-01 1.024e-01 -5.330e+00  
## PolicyTypeSedan - Liability -3.004e+00 2.099e-01 -1.431e+01  
## PolicyTypeSport - All Perils -9.895e+01 1.937e+07 0.000e+00  
## PolicyTypeSport - Collision 5.758e-01 2.889e-01 1.993e+00  
## PolicyTypeSport - Liability -9.960e+01 6.711e+07 0.000e+00  
## PolicyTypeUtility - All Perils -4.112e-01 2.580e-01 -1.594e+00  
## PolicyTypeUtility - Collision -2.539e-01 6.583e-01 -3.860e-01  
## PolicyTypeUtility - Liability -9.944e+01 1.794e+07 0.000e+00  
## VehicleCategorySport NA NA NA  
## VehicleCategoryUtility NA NA NA  
## VehiclePrice30000 to 39000 5.406e-02 1.265e-01 4.270e-01  
## VehiclePrice40000 to 59000 5.598e-01 2.687e-01 2.084e+00  
## VehiclePrice60000 to 69000 5.846e-02 7.770e-01 7.500e-02  
## VehiclePriceless than 20000 7.632e-02 1.575e-01 4.840e-01  
## VehiclePricemore than 69000 1.899e-01 1.762e-01 1.078e+00  
## RepNumber2 -2.186e-02 2.425e-01 -9.000e-02  
## RepNumber3 6.456e-02 2.356e-01 2.740e-01  
## RepNumber4 -2.972e-01 2.544e-01 -1.169e+00  
## RepNumber5 -1.232e-01 2.416e-01 -5.100e-01  
## RepNumber6 -7.044e-03 2.368e-01 -3.000e-02  
## RepNumber7 2.434e-01 2.236e-01 1.088e+00  
## RepNumber8 -3.793e-01 2.575e-01 -1.473e+00  
## RepNumber9 3.522e-02 2.370e-01 1.490e-01  
## RepNumber10 1.078e-01 2.248e-01 4.790e-01  
## RepNumber11 -8.842e-02 2.384e-01 -3.710e-01  
## RepNumber12 -3.325e-01 2.512e-01 -1.324e+00  
## RepNumber13 2.939e-01 2.350e-01 1.251e+00  
## RepNumber14 -3.907e-02 2.453e-01 -1.590e-01  
## RepNumber15 -6.170e-01 2.646e-01 -2.331e+00  
## RepNumber16 -2.576e-01 2.455e-01 -1.049e+00  
## Deductible 3.065e-04 1.041e-03 2.950e-01  
## DriverRating -1.245e-02 3.921e-02 -3.170e-01  
## Days\_Policy\_Accident15 to 30 1.599e+02 2.373e+07 0.000e+00  
## Days\_Policy\_Accident8 to 15 1.602e+02 2.373e+07 0.000e+00  
## Days\_Policy\_Accidentmore than 30 1.612e+02 2.373e+07 0.000e+00  
## Days\_Policy\_Accidentnone 1.616e+02 2.373e+07 0.000e+00  
## Days\_Policy\_Claim8 to 15 -3.749e-01 1.400e+00 -2.680e-01  
## Days\_Policy\_Claimmore than 30 -1.252e+00 8.543e-01 -1.465e+00  
## Days\_Policy\_Claimnone 3.080e+12 1.482e+13 2.080e-01  
## PastNumberOfClaims2 to 4 3.275e-01 1.202e-01 2.726e+00  
## PastNumberOfClaimsmore than 4 1.543e-01 1.842e-01 8.370e-01  
## PastNumberOfClaimsnone 1.648e-01 1.181e-01 1.395e+00  
## AgeOfVehicle3 years 1.858e+00 1.167e+00 1.592e+00  
## AgeOfVehicle4 years 2.352e+00 1.156e+00 2.035e+00  
## AgeOfVehicle5 years 1.946e+00 1.155e+00 1.685e+00  
## AgeOfVehicle6 years 1.833e+00 1.153e+00 1.589e+00  
## AgeOfVehicle7 years 1.608e+00 1.154e+00 1.393e+00  
## AgeOfVehiclemore than 7 1.556e+00 1.159e+00 1.343e+00  
## AgeOfVehiclenew -9.163e-01 2.259e+00 -4.060e-01  
## AgeOfPolicyHolder18 to 20 -2.827e+00 2.319e+00 -1.219e+00  
## AgeOfPolicyHolder21 to 25 -2.813e+00 1.991e+00 -1.413e+00  
## AgeOfPolicyHolder26 to 30 -3.383e+00 1.966e+00 -1.721e+00  
## AgeOfPolicyHolder31 to 35 -3.002e+00 1.955e+00 -1.535e+00  
## AgeOfPolicyHolder36 to 40 -2.955e+00 1.958e+00 -1.509e+00  
## AgeOfPolicyHolder41 to 50 -3.125e+00 1.961e+00 -1.593e+00  
## AgeOfPolicyHolder51 to 65 -3.224e+00 1.965e+00 -1.641e+00  
## AgeOfPolicyHolderover 65 -3.129e+00 1.973e+00 -1.586e+00  
## PoliceReportFiledYes -3.808e-01 3.111e-01 -1.224e+00  
## WitnessPresentYes -2.665e-01 7.547e-01 -3.530e-01  
## AgentTypeInternal -6.812e-01 6.020e-01 -1.132e+00  
## NumberOfSuppliments3 to 5 -4.407e-02 1.687e-01 -2.610e-01  
## NumberOfSupplimentsmore than 5 1.087e-01 1.442e-01 7.540e-01  
## NumberOfSupplimentsnone 1.665e-01 1.298e-01 1.282e+00  
## AddressChange\_Claim2 to 3 years 9.184e-01 4.967e-01 1.849e+00  
## AddressChange\_Claim4 to 8 years -9.505e-01 5.432e-01 -1.750e+00  
## AddressChange\_Claimno change -5.555e-01 4.476e-01 -1.241e+00  
## AddressChange\_Claimunder 6 months 1.034e+02 3.875e+07 0.000e+00  
## NumberOfCars2 vehicles 8.964e-02 5.523e-01 1.620e-01  
## NumberOfCars3 to 4 6.961e-02 2.742e-01 2.540e-01  
## NumberOfCars5 to 8 8.112e-01 1.099e+00 7.380e-01  
## NumberOfCarsmore than 8 -9.918e+01 4.745e+07 0.000e+00  
## Year1995 -1.392e-01 1.038e-01 -1.341e+00  
## Year1996 -1.639e-01 1.103e-01 -1.486e+00  
## BasePolicyCollision NA NA NA  
## BasePolicyLiability NA NA NA  
## Pr(>|z|)   
## (Intercept) 0.83541   
## MonthAug 0.52779   
## MonthDec 0.85345   
## MonthFeb 0.91628   
## MonthJan 0.78594   
## MonthJul 0.38081   
## MonthJun 0.70375   
## MonthMar 0.06660 .   
## MonthMay 0.65738   
## MonthNov 0.78758   
## MonthOct 0.91808   
## MonthSep 0.88024   
## WeekOfMonth2 0.28287   
## WeekOfMonth3 0.90915   
## WeekOfMonth4 0.56215   
## WeekOfMonth5 0.37298   
## DayOfWeekMonday 0.54376   
## DayOfWeekSaturday 0.55974   
## DayOfWeekSunday 0.75836   
## DayOfWeekThursday 0.40992   
## DayOfWeekTuesday 0.48004   
## DayOfWeekWednesday 0.46278   
## MakeBMW < 2e-16 \*\*\*  
## MakeChevrolet 0.03635 \*   
## MakeDodge 0.25513   
## MakeFerrari < 2e-16 \*\*\*  
## MakeFord 0.50970   
## MakeHonda 0.08614 .   
## MakeJaguar < 2e-16 \*\*\*  
## MakeLexus < 2e-16 \*\*\*  
## MakeMazda 0.07555 .   
## MakeMecedes < 2e-16 \*\*\*  
## MakeMercury 0.25345   
## MakeNisson 0.49183   
## MakePontiac 0.03101 \*   
## MakePorche < 2e-16 \*\*\*  
## MakeSaab 0.91280   
## MakeSaturn 0.24197   
## MakeToyota 0.12745   
## MakeVW 0.00859 \*\*   
## AccidentAreaUrban 0.20844   
## DayOfWeekClaimedFriday 0.83541   
## DayOfWeekClaimedMonday 0.83541   
## DayOfWeekClaimedSaturday 0.83541   
## DayOfWeekClaimedSunday 0.83541   
## DayOfWeekClaimedThursday 0.83541   
## DayOfWeekClaimedTuesday 0.83541   
## DayOfWeekClaimedWednesday 0.83541   
## MonthClaimedApr 0.47613   
## MonthClaimedAug 0.08112 .   
## MonthClaimedDec 0.19041   
## MonthClaimedFeb 0.45502   
## MonthClaimedJan 0.80215   
## MonthClaimedJul 0.83412   
## MonthClaimedJun 0.93375   
## MonthClaimedMar 0.39330   
## MonthClaimedMay 0.91199   
## MonthClaimedNov 0.02202 \*   
## MonthClaimedOct 0.72447   
## MonthClaimedSep NA   
## WeekOfMonthClaimed2 0.03786 \*   
## WeekOfMonthClaimed3 0.37364   
## WeekOfMonthClaimed4 0.80580   
## WeekOfMonthClaimed5 0.23786   
## SexMale 0.03039 \*   
## MaritalStatusMarried 0.92773   
## MaritalStatusSingle 0.98155   
## MaritalStatusWidow 0.38148   
## FaultThird Party < 2e-16 \*\*\*  
## PolicyTypeSedan - Collision 9.84e-08 \*\*\*  
## PolicyTypeSedan - Liability < 2e-16 \*\*\*  
## PolicyTypeSport - All Perils 1.00000   
## PolicyTypeSport - Collision 0.04628 \*   
## PolicyTypeSport - Liability 1.00000   
## PolicyTypeUtility - All Perils 0.11092   
## PolicyTypeUtility - Collision 0.69972   
## PolicyTypeUtility - Liability 1.00000   
## VehicleCategorySport NA   
## VehicleCategoryUtility NA   
## VehiclePrice30000 to 39000 0.66922   
## VehiclePrice40000 to 59000 0.03718 \*   
## VehiclePrice60000 to 69000 0.94003   
## VehiclePriceless than 20000 0.62804   
## VehiclePricemore than 69000 0.28115   
## RepNumber2 0.92817   
## RepNumber3 0.78402   
## RepNumber4 0.24257   
## RepNumber5 0.61008   
## RepNumber6 0.97627   
## RepNumber7 0.27642   
## RepNumber8 0.14072   
## RepNumber9 0.88184   
## RepNumber10 0.63163   
## RepNumber11 0.71076   
## RepNumber12 0.18565   
## RepNumber13 0.21110   
## RepNumber14 0.87347   
## RepNumber15 0.01973 \*   
## RepNumber16 0.29409   
## Deductible 0.76836   
## DriverRating 0.75087   
## Days\_Policy\_Accident15 to 30 0.99999   
## Days\_Policy\_Accident8 to 15 0.99999   
## Days\_Policy\_Accidentmore than 30 0.99999   
## Days\_Policy\_Accidentnone 0.99999   
## Days\_Policy\_Claim8 to 15 0.78894   
## Days\_Policy\_Claimmore than 30 0.14285   
## Days\_Policy\_Claimnone 0.83541   
## PastNumberOfClaims2 to 4 0.00642 \*\*   
## PastNumberOfClaimsmore than 4 0.40235   
## PastNumberOfClaimsnone 0.16288   
## AgeOfVehicle3 years 0.11132   
## AgeOfVehicle4 years 0.04184 \*   
## AgeOfVehicle5 years 0.09193 .   
## AgeOfVehicle6 years 0.11206   
## AgeOfVehicle7 years 0.16349   
## AgeOfVehiclemore than 7 0.17939   
## AgeOfVehiclenew 0.68502   
## AgeOfPolicyHolder18 to 20 0.22274   
## AgeOfPolicyHolder21 to 25 0.15770   
## AgeOfPolicyHolder26 to 30 0.08521 .   
## AgeOfPolicyHolder31 to 35 0.12468   
## AgeOfPolicyHolder36 to 40 0.13127   
## AgeOfPolicyHolder41 to 50 0.11110   
## AgeOfPolicyHolder51 to 65 0.10081   
## AgeOfPolicyHolderover 65 0.11278   
## PoliceReportFiledYes 0.22081   
## WitnessPresentYes 0.72405   
## AgentTypeInternal 0.25781   
## NumberOfSuppliments3 to 5 0.79391   
## NumberOfSupplimentsmore than 5 0.45086   
## NumberOfSupplimentsnone 0.19967   
## AddressChange\_Claim2 to 3 years 0.06448 .   
## AddressChange\_Claim4 to 8 years 0.08011 .   
## AddressChange\_Claimno change 0.21456   
## AddressChange\_Claimunder 6 months 1.00000   
## NumberOfCars2 vehicles 0.87106   
## NumberOfCars3 to 4 0.79963   
## NumberOfCars5 to 8 0.46061   
## NumberOfCarsmore than 8 1.00000   
## Year1995 0.17993   
## Year1996 0.13739   
## BasePolicyCollision NA   
## BasePolicyLiability NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4689.6 on 10022 degrees of freedom  
## Residual deviance: 3747.0 on 9884 degrees of freedom  
## AIC: 4025  
##   
## Number of Fisher Scoring iterations: 100

## making predictions and evaluating accuracy of full model

glm\_full\_pred <- predict(glm\_full, newdata = x\_test, type="response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

glm\_full\_pred <-ifelse(glm\_full\_pred >=0.5,1,0)  
  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

glm\_full\_pred <- as.factor(glm\_full\_pred)  
confusionMatrix(data=glm\_full\_pred, y\_test)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 5092 292  
## 1 9 4  
##   
## Accuracy : 0.9442   
## 95% CI : (0.9378, 0.9502)  
## No Information Rate : 0.9452   
## P-Value [Acc > NIR] : 0.6318   
##   
## Kappa : 0.0214   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.99824   
## Specificity : 0.01351   
## Pos Pred Value : 0.94577   
## Neg Pred Value : 0.30769   
## Prevalence : 0.94515   
## Detection Rate : 0.94349   
## Detection Prevalence : 0.99759   
## Balanced Accuracy : 0.50587   
##   
## 'Positive' Class : 0   
##

## Evaluating full model performace using AUC

library(ROCR)

## Loading required package: gplots

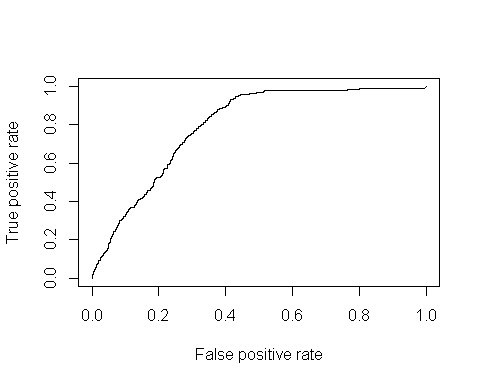
##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

p <- predict(glm\_full, newdata=x\_test, type="response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

pr2 <- prediction(p, y\_test)  
prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")  
plot(prf2)



auc1 <- performance(pr2, measure = "auc")  
auc1 <- auc1@y.values[[1]]  
print(auc1)

## [1] 0.7950657

# Basic Random forest model

library(randomForest)

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

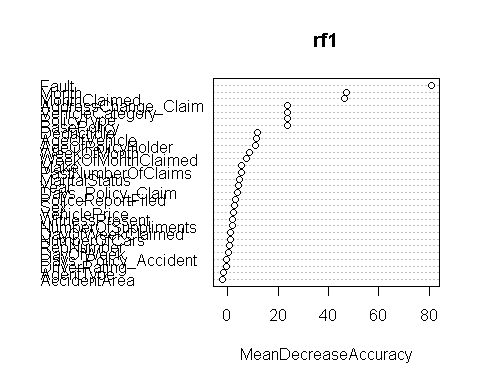
table(claims$FraudFound\_P)

##   
## 0 1   
## 14497 923

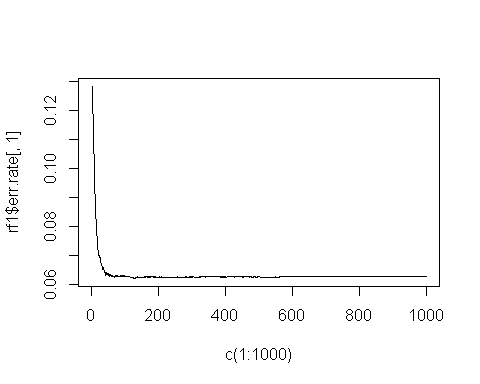
set.seed(999)  
  
rf1 <- randomForest(FraudFound\_P ~ ., data=claims,importance = TRUE, ntree=1000, subset=train)  
rf1

##   
## Call:  
## randomForest(formula = FraudFound\_P ~ ., data = claims, importance = TRUE, ntree = 1000, subset = train)   
## Type of random forest: classification  
## Number of trees: 1000  
## No. of variables tried at each split: 5  
##   
## OOB estimate of error rate: 6.26%  
## Confusion matrix:  
## 0 1 class.error  
## 0 9395 1 0.0001064283  
## 1 626 1 0.9984051037

varImpPlot(rf1, type=1)



plot(c(1:1000), rf1$err.rate[,1], type='l')



rf1\_pred <- predict(rf1, newdata = claims[-train, ])

# RF model accuracy for test set

confusionMatrix(data=rf1\_pred, y\_test)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 5099 294  
## 1 2 2  
##   
## Accuracy : 0.9452   
## 95% CI : (0.9387, 0.9511)  
## No Information Rate : 0.9452   
## P-Value [Acc > NIR] : 0.5155   
##   
## Kappa : 0.0119   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.999608   
## Specificity : 0.006757   
## Pos Pred Value : 0.945485   
## Neg Pred Value : 0.500000   
## Prevalence : 0.945155   
## Detection Rate : 0.944784   
## Detection Prevalence : 0.999259   
## Balanced Accuracy : 0.503182   
##   
## 'Positive' Class : 0   
##

#claims$FraudFound\_P <- as.factor(claims$FraudFound\_P)  
library(caret)  
#data(GermanCredit)  
Train <- createDataPartition(claims$FraudFound\_P, p=0.6, list=FALSE)  
training <- claims[ Train, ]  
testing <- claims[ -Train, ]

# Logistic Regresion with 10 fold cross validation

library(caret)  
# Define training control  
set.seed(123)  
train.control <- trainControl(method = "repeatedcv",   
 number = 10, repeats = 1)  
# Train the model  
model <- train(FraudFound\_P ~., data = claims, method="glm", family="binomial",  
 trControl = train.control)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# Summarize the results  
print(model)

## Generalized Linear Model   
##   
## 15420 samples  
## 30 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 13877, 13879, 13879, 13877, 13878, 13878, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9393648 0.02038686

pred = predict(model, newdata=x\_test)

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

confusionMatrix(data=pred, y\_test)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 5098 289  
## 1 3 7  
##   
## Accuracy : 0.9459   
## 95% CI : (0.9395, 0.9518)  
## No Information Rate : 0.9452   
## P-Value [Acc > NIR] : 0.4204   
##   
## Kappa : 0.0423   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.99941   
## Specificity : 0.02365   
## Pos Pred Value : 0.94635   
## Neg Pred Value : 0.70000   
## Prevalence : 0.94515   
## Detection Rate : 0.94460   
## Detection Prevalence : 0.99815   
## Balanced Accuracy : 0.51153   
##   
## 'Positive' Class : 0   
##

# Logistic Regresion with 10 fold cross validation and upsampling

# Define training control for upsampling  
set.seed(123)  
train.control2 <- trainControl(method = "repeatedcv",   
 number = 10, repeats = 1, sampling = "up")  
# Train the model  
model2 <- train(FraudFound\_P ~., data = claims, method="glm", family="binomial",  
 trControl = train.control2)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# Summarize the results  
print(model2)

## Generalized Linear Model   
##   
## 15420 samples  
## 30 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 1 times)   
## Summary of sample sizes: 13877, 13879, 13879, 13877, 13878, 13878, ...   
## Addtional sampling using up-sampling  
##   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.6598555 0.1398895

pred = predict(model2, newdata=x\_test)

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =  
## ifelse(type == : prediction from a rank-deficient fit may be misleading

confusionMatrix(data=pred, y\_test)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 3270 24  
## 1 1831 272  
##   
## Accuracy : 0.6563   
## 95% CI : (0.6434, 0.669)  
## No Information Rate : 0.9452   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.1445   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.6411   
## Specificity : 0.9189   
## Pos Pred Value : 0.9927   
## Neg Pred Value : 0.1293   
## Prevalence : 0.9452   
## Detection Rate : 0.6059   
## Detection Prevalence : 0.6103   
## Balanced Accuracy : 0.7800   
##   
## 'Positive' Class : 0   
##

# Random Forest with 5 fold cross validation and upsampling

set.seed(123)  
train.control3 <- trainControl(method = "repeatedcv",   
 number = 5, repeats = 1, sampling = "up")  
mtry <- sqrt(ncol(claims))  
#tunegrid <- expand.grid(.mtry=mtry)  
rf\_default <- train(FraudFound\_P~.,   
 data=claims,   
 method='rf',   
 #metric='Accuracy',   
 #tuneGrid=tunegrid,  
 ntree=10 ,  
 trControl=train.control3)  
print(rf\_default)

## Random Forest   
##   
## 15420 samples  
## 30 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold, repeated 1 times)   
## Summary of sample sizes: 12337, 12336, 12335, 12335, 12337   
## Addtional sampling using up-sampling  
##   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.6410508 0.08848772  
## 72 0.9305446 0.12735576  
## 143 0.9240593 0.15302040  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 72.

pred = predict(rf\_default, newdata=x\_test)  
confusionMatrix(data=pred, y\_test)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 5099 0  
## 1 2 296  
##   
## Accuracy : 0.9996   
## 95% CI : (0.9987, 1)  
## No Information Rate : 0.9452   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9964   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.9996   
## Specificity : 1.0000   
## Pos Pred Value : 1.0000   
## Neg Pred Value : 0.9933   
## Prevalence : 0.9452   
## Detection Rate : 0.9448   
## Detection Prevalence : 0.9448   
## Balanced Accuracy : 0.9998   
##   
## 'Positive' Class : 0   
##