LogitPKVSir.R

Santhosh

Sun Jul 02 11:34:13 2017

#Logistic Regression PKV Sir approach  
#Required Libraries   
library(googleVis)

## Creating a generic function for 'toJSON' from package 'jsonlite' in package 'googleVis'

##   
## Welcome to googleVis version 0.6.2  
##   
## Please read Google's Terms of Use  
## before you start using the package:  
## https://developers.google.com/terms/  
##   
## Note, the plot method of googleVis will by default use  
## the standard browser to display its output.  
##   
## See the googleVis package vignettes for more details,  
## or visit http://github.com/mages/googleVis.  
##   
## To suppress this message use:  
## suppressPackageStartupMessages(library(googleVis))

library(corrgram)  
library(Deducer) #ROC curve

## Loading required package: ggplot2

## Loading required package: JGR

## Loading required package: rJava

## Loading required package: JavaGD

## Loading required package: iplots

##   
## Please type JGR() to launch console. Platform specific launchers (.exe and .app) can also be obtained at http://www.rforge.net/JGR/files/.

## Loading required package: car

## Loading required package: MASS

##   
##   
## Note Non-JGR console detected:  
## Deducer is best used from within JGR (http://jgr.markushelbig.org/).  
## To Bring up GUI dialogs, type deducer().

library(sqldf)

## Warning: package 'sqldf' was built under R version 3.4.1

## Loading required package: gsubfn

## Warning: package 'gsubfn' was built under R version 3.4.1

## Loading required package: proto

## Warning: package 'proto' was built under R version 3.4.1

## Loading required package: RSQLite

## Warning: package 'RSQLite' was built under R version 3.4.1

library(lmtest) #log likelihood

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(pscl) #Pseudo r squared

## Loading required package: lattice

## Classes and Methods for R developed in the

## Political Science Computational Laboratory

## Department of Political Science

## Stanford University

## Simon Jackman

## hurdle and zeroinfl functions by Achim Zeileis

#Read the Data from CSV File  
getwd()

## [1] "C:/Home/Work/Data Science/models-and-methods/Logit"

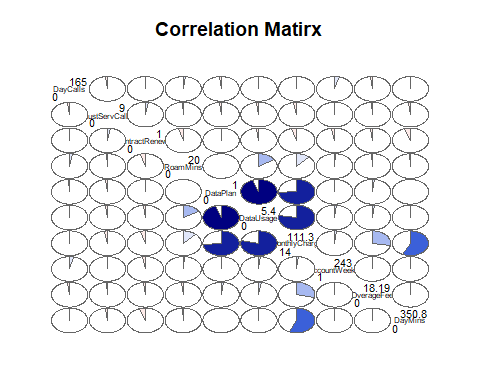
setwd("C:/Home/Work/Data Science/models-and-methods")  
mydata = read.table(file = "Logit/Data/CellPhone.csv", header = T, sep = ",")  
names(mydata)[1] = "Churn"  
sapply(mydata, function(x) summary(x))

## Churn AccountWeeks ContractRenewal DataPlan DataUsage  
## Min. 0.0000000 1.0000 0.0000000 0.0000000 0.0000000  
## 1st Qu. 0.0000000 74.0000 1.0000000 0.0000000 0.0000000  
## Median 0.0000000 101.0000 1.0000000 0.0000000 0.0000000  
## Mean 0.1449145 101.0648 0.9030903 0.2766277 0.8164746  
## 3rd Qu. 0.0000000 127.0000 1.0000000 1.0000000 1.7800000  
## Max. 1.0000000 243.0000 1.0000000 1.0000000 5.4000000  
## CustServCalls DayMins DayCalls MonthlyCharge OverageFee RoamMins  
## Min. 0.000000 0.0000 0.0000 14.00000 0.00000 0.00000  
## 1st Qu. 1.000000 143.7000 87.0000 45.00000 8.33000 8.50000  
## Median 1.000000 179.4000 101.0000 53.50000 10.07000 10.30000  
## Mean 1.562856 179.7751 100.4356 56.30516 10.05149 10.23729  
## 3rd Qu. 2.000000 216.4000 114.0000 66.20000 11.77000 12.10000  
## Max. 9.000000 350.8000 165.0000 111.30000 18.19000 20.00000

#Check for Correlation   
corData = cor(mydata[,-1],method = "pearson")  
corData

## AccountWeeks ContractRenewal DataPlan DataUsage  
## AccountWeeks 1.000000000 -0.024734655 0.002918409 0.014390757  
## ContractRenewal -0.024734655 1.000000000 -0.006006371 -0.019222913  
## DataPlan 0.002918409 -0.006006371 1.000000000 0.945981734  
## DataUsage 0.014390757 -0.019222913 0.945981734 1.000000000  
## CustServCalls -0.003795939 0.024521956 -0.017823944 -0.021722518  
## DayMins 0.006216021 -0.049395824 -0.001684069 0.003175951  
## DayCalls 0.038469882 -0.003754626 -0.011085902 -0.007962079  
## MonthlyCharge 0.012580670 -0.047291399 0.737489653 0.781660429  
## OverageFee -0.006749462 -0.019104644 0.021525559 0.019637372  
## RoamMins 0.009513902 -0.045870743 -0.001317871 0.162745576  
## CustServCalls DayMins DayCalls MonthlyCharge  
## AccountWeeks -0.003795939 0.006216021 0.038469882 0.012580670  
## ContractRenewal 0.024521956 -0.049395824 -0.003754626 -0.047291399  
## DataPlan -0.017823944 -0.001684069 -0.011085902 0.737489653  
## DataUsage -0.021722518 0.003175951 -0.007962079 0.781660429  
## CustServCalls 1.000000000 -0.013423186 -0.018941930 -0.028016853  
## DayMins -0.013423186 1.000000000 0.006750414 0.567967924  
## DayCalls -0.018941930 0.006750414 1.000000000 -0.007963218  
## MonthlyCharge -0.028016853 0.567967924 -0.007963218 1.000000000  
## OverageFee -0.012964219 0.007038214 -0.021448602 0.281766048  
## RoamMins -0.009639680 -0.010154586 0.021564794 0.117432607  
## OverageFee RoamMins  
## AccountWeeks -0.006749462 0.009513902  
## ContractRenewal -0.019104644 -0.045870743  
## DataPlan 0.021525559 -0.001317871  
## DataUsage 0.019637372 0.162745576  
## CustServCalls -0.012964219 -0.009639680  
## DayMins 0.007038214 -0.010154586  
## DayCalls -0.021448602 0.021564794  
## MonthlyCharge 0.281766048 0.117432607  
## OverageFee 1.000000000 -0.011023336  
## RoamMins -0.011023336 1.000000000

corrgram(mydata[,-1], order = TRUE, lower.panel = panel.pie, upper.panel = panel.pie, text.panel = panel.txt, main="Correlation Matirx", diag.panel = panel.minmax)



#Highly correlated variables  
# Data Plan X Data Usage  
# Data Plan X Monthly Charge  
# Monthly Charge X DayMins  
# Monthly Charge X OverageMins  
  
#Performing Linear Regression  
regression = lm(Churn~., data = mydata)  
summary(regression)

##   
## Call:  
## lm(formula = Churn ~ ., data = mydata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.66572 -0.16629 -0.08236 0.02060 1.08844   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.433e-01 5.363e-02 -2.672 0.007580 \*\*   
## AccountWeeks 8.888e-05 1.396e-04 0.637 0.524402   
## ContractRenewal -2.993e-01 1.882e-02 -15.904 < 2e-16 \*\*\*  
## DataPlan -4.175e-02 4.381e-02 -0.953 0.340650   
## DataUsage -2.835e-02 1.933e-01 -0.147 0.883401   
## CustServCalls 5.829e-02 4.222e-03 13.804 < 2e-16 \*\*\*  
## DayMins 1.021e-03 3.272e-03 0.312 0.754936   
## DayCalls 3.409e-04 2.769e-04 1.231 0.218433   
## MonthlyCharge 1.428e-03 1.924e-02 0.074 0.940838   
## OverageFee 1.046e-02 3.280e-02 0.319 0.749780   
## RoamMins 8.765e-03 2.307e-03 3.800 0.000147 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3203 on 3322 degrees of freedom  
## Multiple R-squared: 0.1747, Adjusted R-squared: 0.1722   
## F-statistic: 70.31 on 10 and 3322 DF, p-value: < 2.2e-16

lrPredict = predict(regression)  
df\_lrPredict = as.data.frame(lrPredict)  
sqldf('select count(1) from df\_lrPredict where lrPredict>1 or lrPredict <0 ')

## count(1)  
## 1 474

#There are 474 observations, whose probability is beyond the realistic range of 0 to 1. Hence, we are using Logistic regression, instead of Linear regression.  
  
#Predicting the Churn using Logistic Regression  
#Splitting the data into 70:30  
# 70% for Training  
# 30% for Testing  
set.seed(45)  
splitPropotion = sample(nrow(mydata),nrow(mydata)\*.7)  
mydata\_train = mydata[splitPropotion,]  
mydata\_test = mydata[-splitPropotion,]  
#As we know there are 4 variables that are highly correlated in the following order, we will use them as interaction effect while defining the equation.  
#Highly correlated variables  
# Data Plan X Data Usage  
# Data Plan X Monthly Charge  
# Monthly Charge X DayMins  
# Monthly Charge X OverageMins  
  
  
#Model Equation and Model  
logitModel = glm(Churn~AccountWeeks+ContractRenewal+DataPlan+DataUsage+CustServCalls+DayMins+DayCalls+MonthlyCharge+OverageFee+RoamMins+DataPlan\*DataUsage+DataPlan\*MonthlyCharge+MonthlyCharge\*DayMins+MonthlyCharge\*OverageFee,data = mydata\_train, family = binomial)  
  
#Testing for Goodness of fit  
### Log likelihood Ratio Test  
lrtest(logitModel)

## Likelihood ratio test  
##   
## Model 1: Churn ~ AccountWeeks + ContractRenewal + DataPlan + DataUsage +   
## CustServCalls + DayMins + DayCalls + MonthlyCharge + OverageFee +   
## RoamMins + DataPlan \* DataUsage + DataPlan \* MonthlyCharge +   
## MonthlyCharge \* DayMins + MonthlyCharge \* OverageFee  
## Model 2: Churn ~ 1  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 15 -651.70   
## 2 1 -934.52 -14 565.65 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## The overall test of the model is significant based on the Chisq test and it is highly significant. This tells that likelihood of any customer churning out is heavily dependent on all the variables or at least one variable.  
  
###Pseudo R Square  
pR2(logitModel)

## llh llhNull G2 McFadden r2ML   
## -651.6975412 -934.5213873 565.6476923 0.3026403 0.2153010   
## r2CU   
## 0.3906185

## McFadden RSquare is 30%, which means that 30% of uncertainty of intercept only model (model2 in likelihood ration test) has been explained by full model (model1).  
## goodness of fit is reasonable  
  
#Summary of the model  
summary(logitModel)

##   
## Call:  
## glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +   
## DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +   
## OverageFee + RoamMins + DataPlan \* DataUsage + DataPlan \*   
## MonthlyCharge + MonthlyCharge \* DayMins + MonthlyCharge \*   
## OverageFee, family = binomial, data = mydata\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2884 -0.4239 -0.2787 -0.1816 3.1317   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.932e+00 1.275e+00 3.867 0.00011 \*\*\*  
## AccountWeeks 1.736e-03 1.813e-03 0.958 0.33824   
## ContractRenewal -2.188e+00 1.873e-01 -11.679 < 2e-16 \*\*\*  
## DataPlan 9.061e+00 1.044e+00 8.682 < 2e-16 \*\*\*  
## DataUsage -1.331e+00 2.599e+00 -0.512 0.60859   
## CustServCalls 6.005e-01 5.175e-02 11.603 < 2e-16 \*\*\*  
## DayMins -2.285e-02 4.334e-02 -0.527 0.59801   
## DayCalls 6.276e-04 3.574e-03 0.176 0.86061   
## MonthlyCharge -2.385e-01 2.534e-01 -0.941 0.34675   
## OverageFee -3.353e-01 4.457e-01 -0.752 0.45180   
## RoamMins 8.483e-02 3.010e-02 2.819 0.00482 \*\*   
## DataPlan:DataUsage 3.885e+00 7.599e-01 5.113 3.18e-07 \*\*\*  
## DataPlan:MonthlyCharge -2.250e-01 1.891e-02 -11.899 < 2e-16 \*\*\*  
## DayMins:MonthlyCharge 7.744e-04 7.894e-05 9.810 < 2e-16 \*\*\*  
## MonthlyCharge:OverageFee 9.513e-03 1.946e-03 4.889 1.01e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1869.0 on 2332 degrees of freedom  
## Residual deviance: 1303.4 on 2318 degrees of freedom  
## AIC: 1333.4  
##   
## Number of Fisher Scoring iterations: 6

# Customers who has recently renewed the contract, with significant usage of data during the day, customers’ whose monthly charges are more, and who has paid significantly large amount in past 12 months is negatively impacting the churn, which means they are churning out. Especially recent contact renewals.   
# Customers having data services and also users of roaming services are positively impacting the churn. Especially customers with Data as they are Statistically Significant.   
# Customers who frequently call the call centers seems to be well informed and feel more loyal, which is what reflecting in Positive impact with high Significance.   
# all the correlated variables are statistically very significant and also most of them are positive.  
# Deviance has dropped sharply from Null to Residual, that signifies that Independent variables are impacting the churn strongly  
# Company has to look in the existing contact renewal model to control churn, and also revisit and improvise its data plan so that we can bring in more customers to data and control churn.  
# There is a strong correlation between Data Plan and Usage and Charges, also, significantly impacting the churn. Company might want to look into pricing structure as well.  
  
  
#Odds and Probability  
odds = exp(coef(logitModel))  
prob = odds/(odds + 1)  
odds\_and\_prob = cbind(as.data.frame(odds),as.data.frame(prob)[,1])  
names(odds\_and\_prob)[2] = "prob"  
odds\_and\_prob

## odds prob  
## (Intercept) 138.6586466 0.9928397  
## AccountWeeks 1.0017375 0.5004340  
## ContractRenewal 0.1121867 0.1008704  
## DataPlan 8615.6127404 0.9998839  
## DataUsage 0.2642566 0.2090213  
## CustServCalls 1.8229485 0.6457604  
## DayMins 0.9774059 0.4942869  
## DayCalls 1.0006278 0.5001569  
## MonthlyCharge 0.7878427 0.4406667  
## OverageFee 0.7150881 0.4169396  
## RoamMins 1.0885292 0.5211942  
## DataPlan:DataUsage 48.6640654 0.9798647  
## DataPlan:MonthlyCharge 0.7985558 0.4439983  
## DayMins:MonthlyCharge 1.0007747 0.5001936  
## MonthlyCharge:OverageFee 1.0095580 0.5023781

# If data plan changes by 1 unit, the odds for churn is impacted by 8616 times, compared to the loyal customers.  
# Similar trend we can see for DataPlan with DataUsage  
confint(logitModel)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 2.4045126183 7.4124679570  
## AccountWeeks -0.0018169367 0.0052940193  
## ContractRenewal -2.5563026729 -1.8212032199  
## DataPlan 7.0216934037 11.1263333049  
## DataUsage -6.4345860448 3.7612782618  
## CustServCalls 0.4999006639 0.7029772724  
## DayMins -0.1079669514 0.0620758856  
## DayCalls -0.0063713201 0.0076495736  
## MonthlyCharge -0.7362945513 0.2579601914  
## OverageFee -1.2110122494 0.5374782283  
## RoamMins 0.0263828727 0.1444426560  
## DataPlan:DataUsage 2.4278495847 5.4106117777  
## DataPlan:MonthlyCharge -0.2625842644 -0.1883329602  
## DayMins:MonthlyCharge 0.0006198252 0.0009297892  
## MonthlyCharge:OverageFee 0.0057013234 0.0133350940

# Confident interval also tells the same story  
  
  
# Lets us now Predict the Test data using the model  
logitPredcit = predict(logitModel,newdata = mydata\_test, type = "response")  
df\_logitPredcit = as.data.frame(logitPredcit)  
sqldf('select \* from df\_logitPredcit where logitPredcit > 1 or logitPredcit < 0')

## [1] logitPredcit  
## <0 rows> (or 0-length row.names)

# Since we have converted them to Probability based on Exponentials, we do not see negative probability. This way we have brought all the likelihood with the realistic range.  
  
# Setting the cut-off value.  
ChurnPredicted = as.data.frame(ifelse(logitPredcit>.5,1,0))  
names(ChurnPredicted)[1] = "ChurnPredicted"  
# Testing the Performance or Accuraracy of Fit  
## Confusion matrix or classification table  
glm\_CM=table(mydata\_test$Churn, ChurnPredicted$ChurnPredicted)  
glm\_CM = as.matrix(glm\_CM)  
glm\_TP = glm\_CM[1,1]  
glm\_FN = glm\_CM[1,2]  
glm\_FP = glm\_CM[2,1]  
glm\_TN = glm\_CM[2,2]  
  
# accuracy = (TP+TN)/(TP+FN+FP+TN)  
glm\_accuracy = (glm\_TP + glm\_TN)/(glm\_TP+glm\_TN+glm\_FP+glm\_FN)  
glm\_accuracy

## [1] 0.882

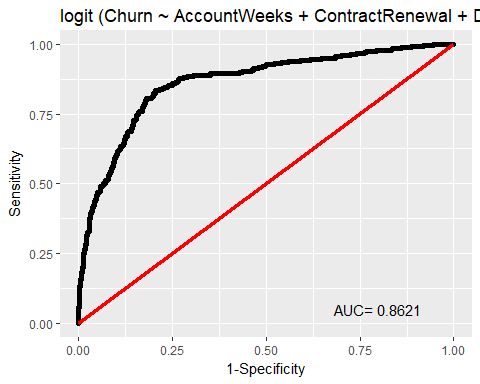
# model is 88% Accurate  
  
# Specificity = TN/(TN+FP)  
glm\_Specificity = glm\_TN/(glm\_TN+glm\_FP)  
glm\_Specificity

## [1] 0.3580247

# 35% of time model predicted churning customers correctly.  
  
# Sensitivity =TP/(TP+FN)  
glm\_Sensitivity = glm\_TP/(glm\_TP+glm\_FN)  
glm\_Sensitivity

## [1] 0.9832936

# 98% of times model predicted non churning customers correctly.  
  
##ROC Curve: Receiver Operating Characteristic(ROC)   
# model’s performance by evaluating the trade offs between true positive rate (sensitivity) and false positive rate(1- specificity).   
# with assumption of cut of point at .5  
#.90-1 = excellent (A)  
#.80-.90 = good (B)  
#.70-.80 = fair (C)  
#.60-.70 = poor (D)  
#.50-.60 = fail (F)  
rocplot(logitModel)



#Here we have AUC - 0.86, which is a good predictive model  
  
# Conclusion  
# using the equation "AccountWeeks+ContractRenewal+DataPlan+DataUsage+CustServCalls+DayMins+DayCalls+MonthlyCharge+OverageFee+RoamMins+DataPlan\*DataUsage+DataPlan\*MonthlyCharge+MonthlyCharge\*DayMins+MonthlyCharge\*OverageFee  
# we should be able to predict the churning customer 86% of time accurately.  
# Organization must look into Data Plan , Usage and Charges to control the churn.