|  |  |  |
| --- | --- | --- |
| setwd("C:/Users/tsraj/Desktop/Acadgild students projects/project2") | | |
|  | library(readxl) | | |
|  | | Churn <- read\_excel("Churn.xls") | |
|  | | View(Churn) | |
|  | | library(plyr) | |
|  | | library(corrplot) | |
|  | | library(ggplot2) | |
|  | | library(gridExtra) | |
|  | | library(ggthemes) | |
|  | | library(caret) | |
|  | | library(MASS) | |
|  | | library(randomForest) | |
|  | | library(party) | |
|  | | library(ggplot2) | |
|  | | library(reshape2) | |
|  | | library(corrplot) | |
|  | | library(e1071) | |
|  | | library(caret) | |
|  | | library(rpart) | |
|  | | library(C50) | |
|  | | library(party) | |
|  | | #library(partykit) | |
|  | | library(randomForest) | |
|  | | library(ROCR) | |
|  | | library(dplyr) | |
|  | | library(car) | |
|  | | str(Churn) | |
|  | | library(Amelia) | |
|  | | any(is.na(Churn)) | |
|  | | # visualize the missing values using the missing map from the Amelia package | |
|  | | missmap(Churn,col=c("yellow","red")) | |
|  | | mydata2<-Churn[,-21] | |
|  | | mydata<-mydata2[,-19] | |
|  | | sapply(mydata, function(x) sum(is.na(x))) | |
|  | | mydata <- mydata[complete.cases(mydata), ] | |
|  | |  | |
|  | |  | |
|  | | intrain<- createDataPartition(mydata$Churn,p=0.8,list=FALSE) | |
|  | | set.seed(2017) | |
|  | | training<- mydata[intrain,] | |
|  | | testing<- mydata[-intrain,] | |
|  | | dim(training); dim(testing) | |
|  | | library (data.table) | |
|  | | library (plyr) | |
|  | | library (stringr) | |
|  | | LogModel <- glm(Churn ~ .,family=binomial(link="logit"),data=training) | |
|  | | print(summary(LogModel)) | |
|  | | anova(LogModel, test="Chisq") | |
|  | | testing$Churn <- as.character(testing$Churn) | |
|  | | testing$Churn[testing$Churn=="No"] <- "0" | |
|  | | testing$Churn[testing$Churn=="Yes"] <- "1" | |
|  | | fitted.results <- predict(LogModel,newdata=testing,type='response') | |
|  | | fitted.results | |
|  | | misClasificError <- mean(fitted.results != testing$Churn) | |
|  | | print(paste('Logistic Regression Accuracy',1-misClasificError)) | |
|  | | # calculating the accuracy rate | |
|  | | accuracyRate <-1-misClasificError | |
|  | | print(accuracyRate) | |
|  | | print("Confusion Matrix for Logistic Regression"); table(testing$Churn, fitted.results > 0.5) | |
|  | | exp(cbind(OR=coef(LogModel), confint(LogModel))) | |
|  | | head(mydata) | |
|  | | summary(mydata) | |
|  | | View(mydata) | |
|  | | sapply(mydata, sd) | |
|  | | cormatrix <- round(cor(mydata), digits = 2 ) | |
|  | | cormatrix | |
|  | | plot.new() | |
|  | | plot(mydata$Churn ~mydata$`Day Mins`) | |
|  | | title('Basic Scatterplot') | |
|  | | ggplot(mydata, aes(x=mydata$`Day Mins`)) + geom\_histogram(binwidth = 1, fill = "white", color = "purple") | |
|  | | #Randomly split data into train and test set | |
|  | | #80% will be ssigned to train set, 20% will be assigned to tst set | |
|  | | barplot(table(mydata$Churn), col= c("green", "red"), main='bar plot of Churn') | |
|  | | text(barplot(table(mydata$Churn), col =c('green' , 'red'), main='bar plot of Churn'), 0,table(mydata$Churn), cex =2 , pos =3) | |
|  | | #proportion | |
|  | | round(prop.table(table(mydata$Churn))\*100,digits = 2) | |
|  | | names(mydata) | |
|  | | normalize<-function(x){return((x-min(x))/(max(x)-min(x)))} | |
|  | | mydata\_n<-as.data.frame(lapply(mydata[1:18],normalize)) | |
|  | | str(mydata) | |
|  | | str(mydata\_n) | |
|  | | mydata\_train<-mydata\_n[1:2666,] | |
|  | | mydata\_test<-mydata\_n[2667:3333,] | |
|  | | mydata\_train\_labels<-mydata\_n[1:2666,7] | |
|  | | mydata\_test\_labels<-mydata\_n[2667:3333,7] | |
|  | | str(mydata\_train) | |
|  | | str(mydata\_train\_labels) | |
|  | | str(mydata\_test) | |
|  | | str(mydata\_test\_labels) | |
|  | | library(class) | |
|  | | #Apply knn | |
|  | | mydata\_test\_pred<-knn(train = mydata\_train,test = mydata\_test, cl=mydata\_train\_labels,k=53) | |
|  | | summary(mydata\_test\_pred) | |
|  | | #Evalulalte model | |
|  | | library(gmodels) | |
|  | | CrossTable(x=mydata\_test\_labels, y=mydata\_test\_pred,prop.chisq = FALSE) | |
|  | |  | |
|  | | sapply(mydata\_n, sd) | |
|  | | cormatrix <- round(cor(mydata\_n), digits = 2 ) | |
|  | | cormatrix | |
|  | | plot.new() | |
|  | | plot(mydata\_n$Churn ~mydata\_n$Day.Mins) | |
|  | | title('Basic Scatterplot') | |
|  | | ggplot(mydata\_n, aes(x=mydata\_n$Day.Mins)) + geom\_histogram(binwidth = 1, fill = "yellow", color = "black") | |
|  | | ggplot(mydata\_n, aes(x=mydata\_n$CustServ.Calls)) + geom\_histogram(binwidth = 1, fill = "green", color = "red") | |
|  | | names(mydata\_n) | |
|  | | #Forward elimination | |
|  | | #Lower AIC indicates a better model | |
|  | | forward <- step(glm(Churn ~ 1, data = mydata\_train), direction = 'forward', scope = ~Account.Length+VMail.Message+Day.Mins + Eve.Mins + | |
|  | | Night.Mins + Intl.Mins + CustServ.Calls + Int.l.Plan + VMail.Plan + | |
|  | | Day.Calls + Day.Charge + Eve.Calls + Eve.Charge + Night.Calls + | |
|  | | Night.Charge + Intl.Calls + Intl.Charge) | |
|  | | logit<- glm(Churn ~Account.Length+Day.Mins+ Day.Charge +CustServ.Calls+VMail.Plan +Eve.Mins+ Eve.Charge+VMail.Message+Day.Calls +Eve.Calls+ Intl.Mins + Night.Calls+Intl.Calls, data = mydata\_train, family = "binomial") | |
|  | | summary(logit) | |
|  | | #evaluate model's fit and performance | |
|  | | influenceIndexPlot(logit, vars = c('Cook', "hat"), id.n =4) | |
|  | | # Confidence interval using log-likelihood | |
|  | | confint(logit) | |
|  | | exp(logit$coefficients) | |
|  | |  | |
|  | | # logistic regression model: | |
|  | | fit <- glm(Churn~.,data =mydata\_train ,family = binomial(link='logit')) | |
|  | | summary(fit) | |
|  | | library(MASS) | |
|  | | step\_fit <- stepAIC(fit,method='backward') | |
|  | | summary(step\_fit) | |
|  | | confint(step\_fit) | |
|  | | #ANOVA on base model | |
|  | | anova(fit,test = 'Chisq') | |
|  | |  | |
|  | | #ANOVA from reduced model after applying the Step AIC | |
|  | | anova(step\_fit,test = 'Chisq') | |
|  | |  | |
|  | | #plot the fitted model | |
|  | | plot(fit$fitted.values) | |
|  | |  | |
|  | | pred\_link <- predict(fit,newdata = mydata\_test,type = 'link') | |
|  | |  | |
|  | | #check for multicollinearity | |
|  | | library(car) | |
|  | | vif(fit) | |
|  | | vif(step\_fit) | |
|  | | pred <- predict(fit,newdata = mydata\_test,type ='response') | |
|  | | #check the AUC curve | |
|  | | library(pROC) | |
|  | | g <- roc( Churn~ pred, data = mydata\_test) | |
|  | | g | |
|  | | plot(g) | |
|  | | library(caret) | |
|  | | #with default prob cut 0.50 | |
|  | | mydata\_test$pred\_Churn <- ifelse(pred<0.8,'yes','no') | |
|  | |  | |
|  | | table(mydata\_test$pred\_Churn,mydata\_test$Churn) | |
|  | |  | |
|  | | #training split of churn classes | |
|  | | round(table(mydata\_train$Churn)/nrow(mydata\_train),2)\*100 | |
|  | | # test split of churn classes | |
|  | | round(table(mydata\_test$Churn)/nrow(mydata\_test),2)\*100 | |
|  | | #predicted split of churn classes | |
|  | | round(table(mydata\_test$pred\_Churn)/nrow(mydata\_test),2)\*100 | |
|  | | #create confusion matrix | |
|  | | #confusionMatrix(mydata\_test$Churn,mydata\_test$pred\_Churn) | |
|  | | #how do we create a cross validation scheme | |
|  | | control <- trainControl(method = 'repeatedcv', | |
|  | | number = 10, | |
|  | | repeats = 3) | |
|  | | seed <-7 | |
|  | | metric <- 'Accuracy' | |
|  | | set.seed(seed) | |
|  | | fit\_default <- train(Churn~., | |
|  | | data = mydata\_train, | |
|  | | method = 'glm', | |
|  | | metric = NaN, | |
|  | | trControl = control) | |
|  | | print(fit\_default) | |
|  | | library(caret) | |
|  | | varImp(step\_fit) | |
|  | | varImp(fit\_default) | |
|  | | library(devtools) | |
|  | | library(woe) | |
|  | |  | |
|  | | library(riv) | |
|  | | iv\_df <- iv.mult(mydata\_train, y="Churn", summary=TRUE, verbose=TRUE) | |
|  | | iv\_df | |
|  | | iv <- iv.mult(mydata\_train, y="Churn", summary=FALSE, verbose=TRUE) | |
|  | | # Plot information value summary | |
|  | |  | |
|  | | iv.plot.summary(iv\_df) | |