**Care for Customer Satisfaction**

Yanan Dai, Lirong Wei

April 2018

**Motivation**

Customer satisfaction is a key for business success in business especially for service industry such as bank, insurance, delivery, etc. However, it’s hard to discovery whether a customer is happy or not as most of them won’t claim their dissatisfaction when they are being serviced. In this project, we are trying to identify dissatisfied customers early in the service process aiming at retaining a good relationship with customers. Doing so would also help companies earn more customers and improve their happiness before it’s too late.

**Methodology/Approach**

The data in our project comes from Kaggle and provided by [Santander Bank](https://www.santanderbank.com/us/personal) which includes hundreds of anonymized numeric features. We built classification models on these data to predict if a customer is satisfied or dissatisfied with their banking experience. The "TARGET" is the variable in the data set to predict. It equals one for unsatisfied customers and 0 for satisfied customers.

First, we used tree-based method: boosting … random forests and classification tree. As there are more than three hundred of variables, it takes a lot of time to compute results of the models, and models may suffer the “curse of dimensionality”. There’s a good property of random forests and boosting is that it can recognize important factors. Then we selected the first 20 important factors concerning the best Gini decrement in random forests. With the 20 variables, we can use other prediction more efficiently. However, the variables selected by random forests maybe correlated with each other which may lead to collinearity problem in models such as logistic regression. Therefore, we further used best subset selection selected the 10 variables. Furthermore, we used logistic regression, support vector machine (SVM) and KNN models. All the detailed codes are in Appendix.

**Results**

The confusion matrix on the test set is shown in Tables. In the random forests, the overall accuracy is 0.9604. However, we are more caring about unsatisfied customers which is denoted by 1, among the 1505 unsatisfied customers we only predict 1 correctly which is very low accuracy, in spite of very high accuracy in predicting satisfied customers.

In the logistic regression, we can predict probability of unsatisfied customers. They we can control threshold to help us predict more unsatisfied customers. For example, when we use threshold of 0.1, we can predict more unsatisfied customers correctly with sacrifice of lower accuracy in predicting unsatisfied customers. The ROC curve of logistic regression in this case is shown in Figure 1. Companies can select threshold according to their need. AUC of the logistic regression model is 0.7607. We further used SVM and KNN, and we can compare models according to their AUC which indicates with same predictors logistics regression performs best on the test set in this case. However, when we use boosting on all the 370 predictors, though it takes time to compute but it has higher

|  |  |  |  |
| --- | --- | --- | --- |
| Logistic regression  (p>0.04) | | True | |
| 0 | 1 |
| Predicted | 0 | 26450 | 464 |
| 1 | 10055 | 1041 |

|  |  |  |  |
| --- | --- | --- | --- |
| Logistic regression  (p>0.1) | | True | |
| 0 | 1 |
| Predicted | 0 | 35038 | 1300 |
| 1 | 1467 | 205 |

AUC than logistic regression.

|  |  |  |  |
| --- | --- | --- | --- |
| Random Forests | | True | |
| 0 | 1 |
| Predicted | 0 | 36503 | 1504 |
| 1 | 2 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| KNN | | True | |
| 0 | 1 |
| Predicted | 0 | 35481 | 1409 |
| 1 | 21024 | 96 |

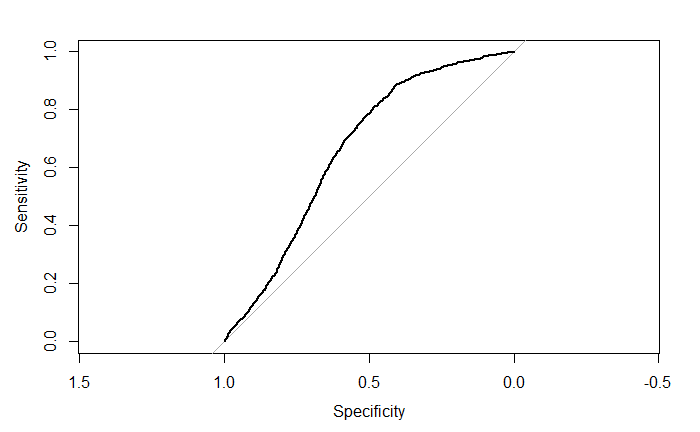
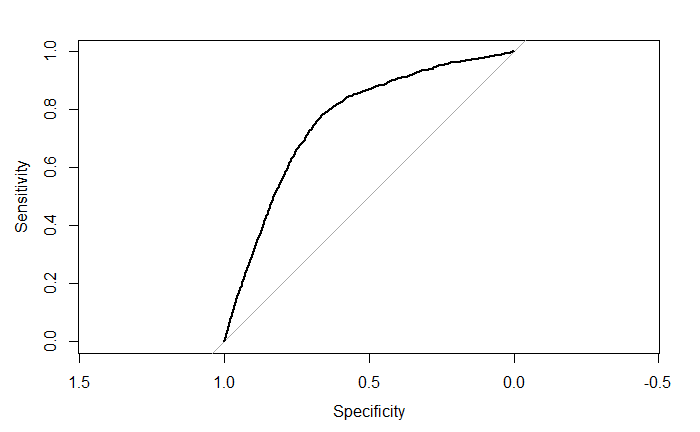


Figure 1. ROC curve of logistic regression (left) ROC curve of SVM regression (right)

|  |  |
| --- | --- |
| methods | AUC |
| boosting |  |
| Logistic regression | 0.7607 |
| SVM | 0.6601 |
| KNN | 0.5186 |

**Conclusion**

We built several models for predicting customer’s satisfaction and we see that by comparing ROC curve and AUC of different models, boosting and logistic regression have the best results. For companies, they are targeting unsatisfied customers to improve their service and expand market. They can select threshold on the ROC curve to control the accuracy of detecting more unsatisfied customers while keep a relative high accuracy of recognizing satisfied customers.

Appendix

## 0. background

We all know that customer satisfaction or loyal customer is one of the key guarantees of successful business especially for service industry. Recognizing unsatisfied customers is important for all kinds of service industry like banking. The question related with the dataset in this proposal is proposed by Santander Bank who aims to identify their unsatisfied customers in the early stage of building the customer relationships. With recognition of unhappy customers, they can specify targets and improve the customer experience to retain the unhappy ones. We will investigate whether bank customers are satisfied with their banking experience or not, based on a range of features. The data can be downloaded in csv format from <https://www.kaggle.com/c/santander-customer-satisfaction/data>. The dataset is from Santander Bank. There are 76020 observations on 370 features and 1 response. The challenge of thie data set is that it is high dimensional and the class in response is high unblanced, we attemped different methods to deal with it.

**Response:** “TARGET” is the response variable which is categorical with 2 classes. One for unsatisfied customer and zero for satisfied customer. **Features:** There are 370 anonymized features including customer’s ID. Most features are numeric variables. As there are so many features with customers, we need to explore data to identify important features before classification.

## 1. Prepare Data

satisf<- read.csv("train.csv", header = TRUE)  
#There are 370 predictors and 1 response variable  
dim(satisf)  
#Most predictors are numberic,all of them are anoymous  
str(satisf,list.len = 100)   
#The response variable  
names(satisf)[371]   
# The satisfied customers and unsatisfied cutomers are unbalanced  
# There are 96 percent of satisfied customers which 4% unsatisfied  
prop.table(table(satisf$TARGET))\*100

## 2. Tree based method

# Split it into train and test  
set.seed(1)  
train<-sample(c(1:nrow(satisf)),nrow(satisf)/2)  
satisf.train<-satisf[train,]

#Use classification  
library(tree)  
  
satisf.train$TARGET<-as.factor(satisf.train$TARGET)  
tree.satisfaction<-tree(TARGET~.-ID,data=satisf.train)  
summary(tree.satisfaction)  
plot(tree.satisfaction)  
text(tree.satisfaction,pretty = 0)  
tree.satisfaction  
  
#predict  
satisf.test<-satisf[-train,]  
satisf.test$TARGET<-as.factor(satisf.test$TARGET)  
tree.pred=predict(tree.satisfaction,satisf.test,type="class")  
table(tree.pred,satisf.test$TARGET)

As we can see the tree can only predict zero value because of high unbalance in response value. I may fail to predict satisfied customers. Then we’d like to use random forests.

library(randomForest)  
set.seed(1)  
rf.satisfaction<-randomForest(TARGET~.-ID,data = satisf.train,ntree=30,importance=TRUE)  
importance(rf.satisfaction)  
  
#Variable Importance  
var.imp = data.frame(importance(rf.satisfaction,   
 type=2))  
var.imp$Variables = row.names(var.imp)   
print(var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),])  
  
#predict  
rf.pred<-predict(rf.satisfaction,newdata = satisf.test)  
table(rf.pred,satisf.test$TARGET)

As we can see random forest performs better than individual tree of course. The result’s specificity is very high while sensitivity is very low. However, we are more care about true positives which indicates accuracy of predicting unsatisfied customers. Acutally, random forest can be used to select a subset of variables that are related to the response. As we know “curse of dimensionality” may greatly influence our analysis, and efficient of computing, therefore, we choose to only use thoes important predictors selected by random forest.

#only use the first 20 important varaibles  
attach(satisf)  
subSatisfy<-satisf[,c("TARGET","var15","var38","saldo\_var30","saldo\_var42","saldo\_medio\_var5\_ult3","saldo\_medio\_var5\_hace2","saldo\_medio\_var5\_hace3","saldo\_medio\_var5\_ult1","saldo\_var5","num\_var45\_ult3","num\_var45\_hace3","num\_var45\_hace2","num\_var22\_ult3","num\_med\_var45\_ult3","num\_var45\_ult1","num\_var22\_hace2","imp\_op\_var41\_ult1","imp\_op\_var39\_ult1","num\_var22\_hace3","num\_var22\_ult1")]  
subSatisfy.train<-subSatisfy[train,]

## 3. Logistic regression

Then we used logistics regression, however, as selected variables by random forests can’t avoid correaltion between factors which may lead to colliearty and influence results of logstic regression.Therefore, we examined correlation between predictors:

cor(subSatisfy)

summary(subSatisfy)

As we can see from the correlation matrix, many variables are high correlated, so then we used best subset selection:

library(leaps)  
regfit.full=regsubsets(TARGET~.,subSatisfy,nvmax=20)  
reg.summary=summary(regfit.full)  
par(mforw=c(2,2))  
plot(reg.summary$rss,xlab="Number of Variables",ylab="RSS",type="b")  
plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="b")  
plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type="b")  
plot(reg.summary$bic,xlab="Number of Variables",ylab="bic",type="b")  
points(10,reg.summary$bic[10],col="red",cex=2,pch=20)

From the plot of RSS, adjusted , and BIC we can decide using model with 10 variables have the best measure performance.

coef(regfit.full,10)

Now, we can use the ten variables to do prediction using logistics regression.

#fit logistic regression model  
subSatisfy.train$TARGET=as.factor(subSatisfy.train$TARGET)  
glm.fits<-glm(TARGET~var15+var38+saldo\_var30+saldo\_var42+saldo\_var5+num\_var45\_ult3+  
 num\_med\_var45\_ult3+imp\_op\_var39\_ult1+  
 num\_var45\_ult1+num\_var22\_ult1,  
 data=subSatisfy.train,family = binomial)  
summary(glm.fits)  
  
#predict  
subSatisfy.test=subSatisfy[-train,]  
subSatisfy.test$TARGET=as.factor(subSatisfy.test$TARGET)  
glm.probs=predict(glm.fits,type="response",newdata = subSatisfy.test)

We need to set threshold for logistics regression result to decide the class.

#make sure 1 is for 1 as dummy variable  
contrasts(subSatisfy.train$TARGET)  
glm.pred=rep(0,nrow(subSatisfy.test))  
#0.5 threshold  
glm.pred[glm.probs>0.5]=1  
table(glm.pred,subSatisfy.test$TARGET)

As we can see when threshold is 0.5, it can’t predicted 1 correctly at all. So we decrease the threshold in order to improve prediction of the unisatisfied (1) customers.

glm.pred=rep(0,nrow(subSatisfy.test))  
#0.1 threshold  
glm.pred[glm.probs>0.1]=1  
table(glm.pred,subSatisfy.test$TARGET)  
#0.04 threshould  
glm.pred=rep(0,nrow(subSatisfy.test))  
glm.pred[glm.probs>0.04]=1  
table(glm.pred,subSatisfy.test$TARGET)

Though now we can predict more unsatisfied customers (1041/1041+464) correctly but now the false positive is so high. We can draw the roc curve of the model.

subSatisfy.test$prob=glm.probs  
library(pROC)  
g <- roc(TARGET ~ prob, data = subSatisfy.test)  
plot(g)   
auc(g)

## 4. Support vector machine

First, we use the linear kernel.

dat=satisf[,c("TARGET","var15","var38","saldo\_var30","saldo\_var42","saldo\_var5","num\_var45\_ult3","num\_med\_var45\_ult3","imp\_op\_var39\_ult1","num\_var45\_ult1","num\_var22\_ult1")]  
dat$TARGET=as.factor(dat$TARGET)  
library(e1071)  
#cross-validation to find the best cost  
set.seed(1)  
tune.out=tune(svm,TARGET~.,data=dat[train,],kenel="linear",ranges=list(cost=c(0.001,0.01,0.1,1,2,3,4,5,10,20,50,80,100)))  
bestmod=tune.out$best.model  
summary(bestmod)  
#predict with best model  
ypred=predict(bestmod,dat[-train,])  
table(predict=ypred,truth=dat[-train,]$TARGET)

We can see that linear model can’t predict unsitisfied customers(1), then we try non-linear boundary.

#use c-v to decide cost and gamma  
set.seed(2)  
tune.out=tune(svm,TARGET~.,data=dat[train,],kenel="radial",ranges=list(cost=c(0.001,0.01,0.1,5,10,50,100)),gamma=c(0.1,1,2,3,4))  
bestmod=tune.out$best.model  
summary(bestmod)  
#predict  
table(true=dat[-train,]$TARGET,predict=predict(bestmod,newdata=dat[-train,]))

It seems that SVM is not suitable in this case, it not able to predict unsatisfied customers. We can look at ROC curve and AUC of SVM model. We can see that it is not as good as logstic regression.

predict=predict(bestmod,newdata=dat[-train,])  
subSatisfy.test$prob=predict  
g <- roc(TARGET ~ prob, data = subSatisfy.test)  
plot(g)   
auc(g)

Finnaly, we used KNN.

library(class)  
#when k=1  
knn.pred=knn(train=dat[train,-1],test=dat[-train,-1],dat[train,1],k=1)  
table(knn.pred,dat[-train,1])  
#when k=2  
knn.pred=knn(train=dat[train,-1],test=dat[-train,-1],dat[train,1],k=2)  
table(knn.pred,dat[-train,1])

We can see that K=1 are doing good in predicting unsatisfied customers. But mays be that that result to overfit to the data. Then we repeat using K=2.We can also plot ROC curve and AUC of KNN which is very low.

knn.probability=knn(train=dat[train,-1],test=dat[-train,-1],dat[train,1],k=2,prob=TRUE)  
subSatisfy.test$prob=as.numeric(knn.probability)  
g <- roc(TARGET ~ prob, data = subSatisfy.test)  
plot(g)   
auc(g)

## 5. Boosting

Code paste here!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1