**Marketing Analytics of retaining**

**customers at Pilgrim Bank**

1. **Executive Summary**

What does a bank care about most? Where do Banks’ most fundamental interests come from? The answer, no doubt, is the customer. In this project, to analyze how the channel use can tell if a customer will stay with a bank in the next year thus to develop a plan to retain customers, we used a set of data which contains 31634 observations from Pilgrim Bank. We have conducted statistical modeling to analyze and extract relevant information that has an impact on retaining customer.

1. **Purpose Statement**

Customer loss is an important indicator for Banks. Getting new customers costs around five times as much as maintaining old ones. Therefore, it is important to excavate convection information from massive user data and establish an efficient customer retaining program. It is necessary to build an evaluation model to reasonably analyze which factors within bank’s control can affect customer retention after adjustment. For example, based on the factors like geographical region, transaction time, gender and the income, we can provide different maintenance methods for different customers, so as to reduce the customer loss rate.

1. **Basic Description of Statistical Methodology**

In this project, because predicting customer retention is a binary statistical problem, we used logistic regression model to eliminate relationships between factors and dependent variable. The stepwise function determined that the variables that have the most significant predictive ability are "age", "income", "tenure", "district","profit15", "online15" and "billpay15". In addition, we also use many machine learning methods to ensure the accuracy of the model. According to the AUC results, neural network and logistic regression outperform the others.

1. **Analysis of Data Findings**
2. **Clean and subset data**

Once we get the data, we are supposed to be clear about what the data look like and what the characteristics of the data are, which are critical for subsequent analysis. After inputting the data set, we can see the dimension of the data by using **str()** and **dim()** function in R. There are **31634** observations and **9** variables, among which we add a new column called “group” into the data frame. (*Appendix A, tab1*) For ease of modeling, we redefined the variables like "retain", "age", "income", "tenure", "district","profit15", "online15" and "billpay15". (*Appendix A, tab2*) In this part, the new set of data could help us reduce the number of levels of every predictor, which can help us classify individuals into groups easily.

1. **Build model and find out important predictors**

We put all the variables we redefined into the logistic regression model. We can see all of them are significant except the tenure from anova() function in R. Then we use step() function and find out that the best model is involved with all of the variables. (*Appendix B, Fig5*) This proves that each variable has a significant contribution to the results of the model, which also means that the user’s personal information and their behavior affect the retention rate.

However, we cannot make a plan for every variable, so we have to find out the variables that are more meaningful. In order to find out the most important predictors, we turn to random forest method and use varImpPlot() to plot out the Accuracy and Gini index. The reason why we use random forest model is that the accuracy and reliability of the model are further improved on the basis of decision tree.

The result is **"profit15","income"** are the most effective features among them. **"age"** also plays an important role there. That is to say, the demographics of a customer kind of help predict the retention. However, the factors like **"online15", "billpay15"** seem to have no big impacts on whether customers will stay with the bank. And except **"age"**, the other factorsfrom demographic information like **"district"** is not that important for the retention prediction. (*Appendix B, Fig1*)

1. **Build five different models and choose the best one**

Using only one classification model does not determine that this model matches the data set best. We need to do more attempts. So we selected several popular methods in the classifiers. We put data into train and test set. Through modeling by train data set, we can get the best parameters for every model. (*Appendix B, Fig2*) We can see from the figure that the area of curve in *Neural Network, Logistic Regression and Naïve Bayes* seems better than others. That is to say, for this data set, those three models fit the customers best.

To ensure the accuracy of the classifier, we need to figure out the most outperformed one by calculating the **AUC** value.(The ROC curve can reflect the classification effect of classifier, but it is not intuitive enough. It is the area under the curve, and it intuitively reflects the classification ability of the curve expression.)Within the five different methods, we can tell that among the training set, Neural Network outperforms the others which holds AUC value as **0.6549053**, but the logistic regression did better among the test data set which has the value of **0.6447815**. (*Appendix B, Fig3*)

1. **Make predictions about individual customers**

Now that we have the best model, we can make predictions about the individual customers.(in this part, we choose neural network for its best performance among the training models) The information we have is a 22 year old guy living in Northeast Texas who earns $65,000/year without any history of this bank. After model prediction, we found that he has a probability of **68.8%** to join this bank, and the binary result is **"Pilgrim Bank"**. (*Appendix B, Fig4*) This prediction tells us that we still have a great chance to get new customers, and we need to consider this as important, so that we can not only retain old users, but also attract new users, which can keep the banks’ customers flow in the long run.

1. **Conclusions and Managerial Recommendations**

The example above gives us a hint that age, income and district are very important for the retention. Combined with the result of the important predictors, we can think of online payment, age, geographical location and income as very important factors.

In response to this result, we have come up with a set of measures which are suitable for this situation.

1. **Payment online**

For get a better frequency of usage of bills online, we need to improve the user experience of online payment. In addition, we are supposed to use user-inspired policies, like ole users who recommend new users can get coupons or dollars. This aims to assure the activity of online payment of users.

1. **Age**

We can offer different policies for different aged groups. We can achieve personalized recommendation about our bank products and special offers. This will not only improve customers’ feelings, but also make them feel cared.

1. **Location**

For different places, especially where we can get or retain customers more easily, we can carry out some activities adapted to local culture in different districts. This is also a way to make customer feel cared.

1. **Income**

Targeted at low-income group, we can make some preferential policies to attract them to stay with the bank. Targeted at high-income group, we should pay more attention to the management of their private user accounts. In addition, we can also introduce some financial products targeted at high-income people.

1. **New customer**

For new users, we should formulate some preferential activities to quickly occupy the market. Although it is very difficult to obtain new users, according to our prediction above, it is not impossible to achieve this goal. So we should work hard on new user preferential policies, user experience and valuable investment products to win the attention of new customers.

1. **Appendix**
2. **Tables**

**(1)**

**(2)**

1. **Figures**

**(1)**

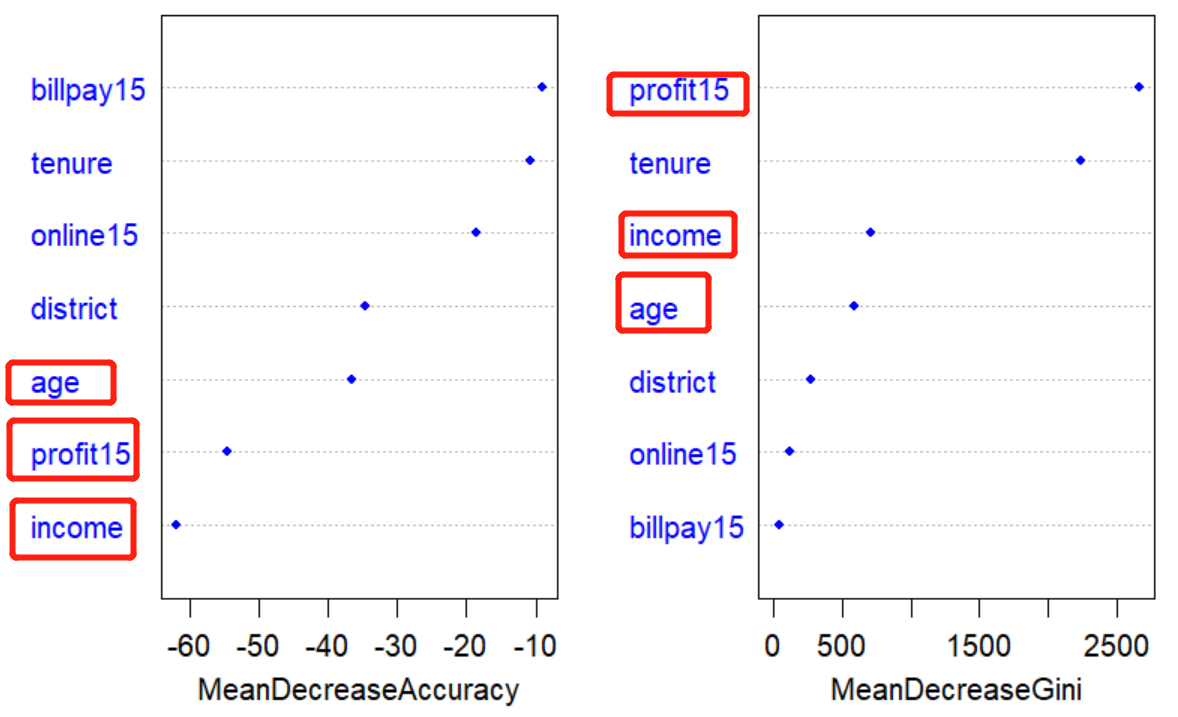


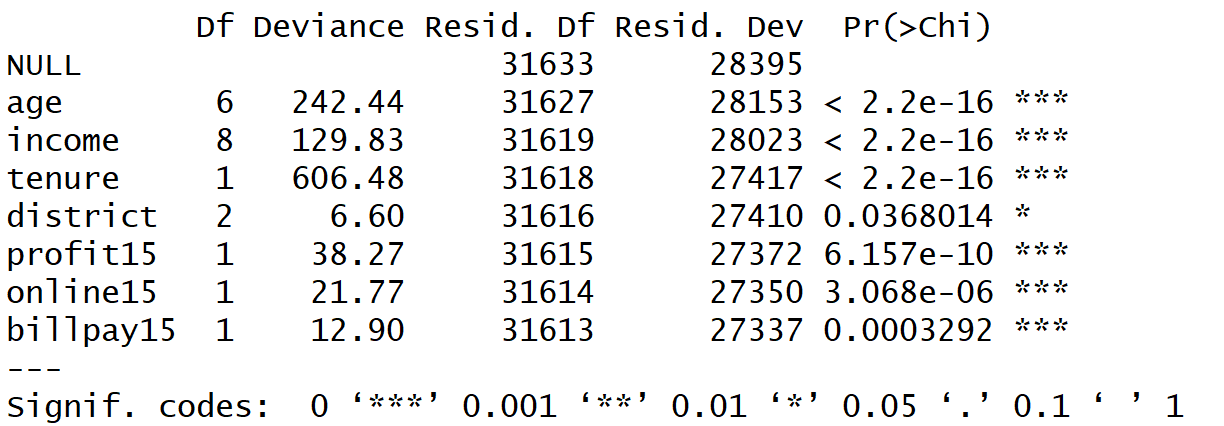
Fig1 Variable Importance

**(2)**

**(3)**

**(4)**

**(5)**

****

**Fig5 The output of ANOVA**

1. **R code**
2. **Input the data**

setwd("C:/STONY/Practice/R(No.4)")

#install.packages("ROCR")

#install.packages("e1071")

#install.packages("randomForest")

#install.packages("nnet")

library(lattice) # lattice plot

library(vcd) # mosaic plots

library(e1071) # support vector machines

library(randomForest) # random forests

library(nnet) # neural networks

library(ROCR) # ROC curve objects for binary classification

# user-defined function for plotting ROC curve using ROC objects from ROCR

**plot.roc <- function(train.roc,train.auc,test.roc,test.auc) {**

**plot(train.roc,col="blue",lty="solid",main="",lwd=2,**

**xlab="False Positive Rate",ylab="True Positive Rate")**

**plot(test.roc,col="red",lty="dashed",lwd=2,add=TRUE)**

**abline(c(0,1))**

**train.legend <- paste("Training AUC = ",round(train.auc,digits=3))**

**test.legend <- paste("Test AUC = ",round(test.auc,digits=3))**

**legend("bottomright",legend=c(train.legend,test.legend),**

**lty=c("solid","dashed"),lwd=2,col=c("blue","red"))**

**}**

# rea the data

**banking.df <- read.csv("online\_banking.csv")**

**str(banking.df)**

**dim(banking.df)**

**head(banking.df)**

**summary(banking.df)**

# redefine the data with levels

**banking.df$age.cat <- ifelse(banking.df$age==1,"<15",**

**ifelse(banking.df$age==2,"15-24",**

**ifelse(banking.df$age==3,"25-34",**

**ifelse(banking.df$age==4,"35-44",**

**ifelse(banking.df$age==5,"45-54",**

**ifelse(banking.df$age==6,"55-64",">65"))))))**

**banking.df$age.cat <- factor(banking.df$age.cat,**

**level=c("<15","15-24","25-34","35-44","45-54","55-64",">65"))**

**banking.df$income.cat <- ifelse(banking.df$income==1,"<15K",**

**ifelse(banking.df$income==2,"15K-20K",**

**ifelse(banking.df$income==3,"20K-30K",**

**ifelse(banking.df$income==4,"30K-40K",**

**ifelse(banking.df$income==5,"40K-50K",**

**ifelse(banking.df$income==6,"50K-75K",**

**ifelse(banking.df$income==7,"75K-100K",**

**ifelse(banking.df$income==8,"100K-125K",">125K"))))))))**

**banking.df$income.cat <- factor(banking.df$income.cat,**

**levels=c("<15K","15K-20K","20K-30K","30K-40K","40K-50K","50K-75K","75K-100K","100K-125K",">125K"))**

**banking.df$district <- factor(ifelse(banking.df$district==1,"Texas Panhandle",**

**ifelse(banking.df$district==2,"Northeast Texas","North Central Texas")),**

**levels=c("Texas Panhandle","Northeast Texas","North Central Texas"))**

**banking.df$online15.cat <- ifelse(banking.df$online15==0,"No","Yes")**

**banking.df$online15.cat <- factor(banking.df$online15.cat,levels=c("No","Yes"))**

**banking.df$billpay15.cat <- ifelse(banking.df$billpay15==0,"No","Yes")**

**banking.df$billpay15.cat <- factor(banking.df$billpay15.cat,levels=c("No","Yes"))**

**banking.df$retain <- ifelse(is.na(banking.df$profit16)==FALSE,"Yes","No")**

**banking.df$retain <- factor(banking.df$retain,levels=c("No","Yes"),**

**labels=c("Other Bank","Pilgrim Bank"))**

**str(banking.df)**

**dim(banking.df)**

**head(banking.df)**

**summary(banking.df,maxsum=10)**

**banking.use.df <- banking.df[,c(15,11,12,3:5,13,14)]**

**colnames(banking.use.df) <- c("retain","age","income","tenure","district",**

**"profit15","online15","billpay15")**

**str(banking.use.df)**

**dim(banking.use.df)**

**head(banking.use.df)**

**summary(banking.use.df,maxsum=10)**

**bk.df <- banking.use.df**

1. **Variable Importance**

# !! THIS PART HELPS WITH QUESTION 1 !!

**model <- {retain~age+income+tenure+district+profit15+online15+billpay15}**

**fit <- glm(model,family=binomial,data=bk.df)**

**summary(fit)**

**anova(fit,test="Chisq")**

**step(fit,direction="both")**

**set.seed(1234)**

**bk.rf.fit <- randomForest(model,data=bk.df,mtry=3,ntree=1000,importance=TRUE)**

**varImpPlot(bk.rf.fit,color="blue",pch=20,cex=1.25,main="")**

**# importance(bk.rf.fit)**

1. **Training-and-Test Regimen**

# !! THIS PART HELPS WITH QUESTION 2 !!

**set.seed(1234)**

**partition <- sample(nrow(bk.df),replace=FALSE)**

**bk.df$group <- ifelse(partition<(2/3)\*nrow(bk.df),1,2)**

**bk.df$group <- factor(bk.df$group,levels=c(1,2),labels=c("TRAIN","TEST"))**

**bk.train.df <- subset(bk.df,subset=(group=="TRAIN"),**

**select=c("retain","age","income","tenure","district",**

**"profit15","online15","billpay15"))**

**bk.test.df <- subset(bk.df,subset=(group=="TEST"),**

**select=c("retain","age","income","tenure","district",**

**"profit15","online15","billpay15"))**

**bk.train.df <- na.omit(bk.train.df)**

**bk.test.df <- na.omit(bk.test.df)**

**if(length(intersect(rownames(bk.train.df),rownames(bk.test.df)))!= 0) {**

**print("\nProblem with partition")**

**}**

1. **Model Calibration and Model Validation**

# !! THIS PART HELPS WITH QUESTION 2 !!

**##### [1] LOGISTIC REGRESSION #####**

**bk.train.lr.fit <- glm(model,family=binomial,data=bk.train.df)**

**# area under ROC curve for TRAINING data**

**bk.train.df$lr.predprob <- predict(bk.train.lr.fit,type="response")**

**bk.train.lr.pred <- prediction(bk.train.df$lr.predprob,bk.train.df$retain)**

**bk.train.lr.auc <- as.numeric(performance(bk.train.lr.pred,"auc")@y.values)**

**# area under ROC curve for TEST data**

**bk.test.df$lr.predprob <- as.numeric(predict(bk.train.lr.fit,**

**newdata=bk.test.df,type="response"))**

**bk.test.lr.pred <- prediction(bk.test.df$lr.predprob,bk.test.df$retain)**

**bk.test.lr.auc <- as.numeric(performance(bk.test.lr.pred,"auc")@y.values)**

**# ROC for logistic regression**

**bk.train.lr.roc <- performance(bk.train.lr.pred,"tpr","fpr")**

**bk.test.lr.roc <- performance(bk.test.lr.pred,"tpr","fpr")**

**plot.roc(train.roc=bk.train.lr.roc,train.auc=bk.train.lr.auc,**

**test.roc=bk.test.lr.roc,test.auc=bk.test.lr.auc)**

**##### [2] SUPPORT VECTOR MACHINES #####**

**# set.seed(1234)**

**# bk.trainsamp <- sample(1:dim(bk.train.df)[1],size=1000,replace=FALSE)**

**# bk.trainsamp.df <- bk.train.df[bk.trainsamp,]**

**# rownames(bk.trainsamp.df) <- NULL**

**# set.seed(1234)**

**# bk.train.svm.tune <- tune(svm,model,data=bk.trainsamp.df,**

**# ranges=list(gamma=2^(-16:1),cost=2^(0:8)),**

**# tunecontrol=tune.control(sampling="fix"))**

**# set.seed(1234)**

**# bk.train.svm.fit <- svm(model,data=bk.train.df,**

**# cost=bk.train.svm.tune$best.parameters$cost,**

**# gamma=bk.train.svm.tune$best.parameters$gamma,**

**# probability=TRUE)**

**set.seed(1234)**

**bk.train.svm.fit <- svm(model,data=bk.train.df,cost=1,gamma=1.525879e-05,**

**probability=TRUE)**

**# area under ROC curve for TRAINING data**

**bk.train.svm.predict <- predict(bk.train.svm.fit,bk.train.df,probability=TRUE)**

**bk.train.df$svm.predprob <- attr(bk.train.svm.predict,"probabilities")[,1]**

**bk.train.svm.prediction <- prediction(bk.train.df$svm.predprob,bk.train.df$retain)**

**bk.train.svm.auc <- as.numeric(performance(bk.train.svm.prediction,"auc")@y.values)**

**# area under ROC curve for TEST data**

**bk.test.svm.predict <- predict(bk.train.svm.fit,bk.test.df,probability=TRUE)**

**bk.test.df$svm.predprob <- attr(bk.test.svm.predict,"probabilities")[,1]**

**bk.test.svm.prediction <- prediction(bk.test.df$svm.predprob,bk.test.df$retain)**

**bk.test.svm.auc <- as.numeric(performance(bk.test.svm.prediction,"auc")@y.values)**

**# ROC for support vector machines classification**

**bk.train.svm.roc <- performance(bk.train.svm.prediction,"tpr","fpr")**

**bk.test.svm.roc <- performance(bk.test.svm.prediction,"tpr","fpr")**

**plot.roc(train.roc=bk.train.svm.roc,train.auc=bk.train.svm.auc,**

**test.roc=bk.test.svm.roc,test.auc=bk.test.svm.auc)**

**##### [3] NEURAL NETWORKS #####**

**set.seed(1234)**

**bk.train.nnet.fit <- nnet(model,data=bk.train.df,size=3,decay=0,**

**probability=TRUE,trace=FALSE)**

**# area under ROC curve for TRAINING data**

**bk.train.df$nnet.predprob <- as.numeric(predict(bk.train.nnet.fit,newdata=bk.train.df))**

**bk.train.nnet.prediction <- prediction(bk.train.df$nnet.predprob,bk.train.df$retain)**

**bk.train.nnet.auc <- as.numeric(performance(bk.train.nnet.prediction,"auc")@y.values)**

**# area under ROC curve for TEST data**

**bk.test.df$nnet.predprob <- as.numeric(predict(bk.train.nnet.fit,newdata=bk.test.df))**

**bk.test.nnet.prediction <- prediction(bk.test.df$nnet.predprob,bk.test.df$retain)**

**bk.test.nnet.auc <- as.numeric(performance(bk.test.nnet.prediction,"auc")@y.values)**

**# ROC for neural network classification**

**bk.train.nnet.roc <- performance(bk.train.nnet.prediction,"tpr","fpr")**

**bk.test.nnet.roc <- performance(bk.test.nnet.prediction,"tpr","fpr")**

**plot.roc(train.roc=bk.train.nnet.roc,train.auc=bk.train.nnet.auc,**

**test.roc=bk.test.nnet.roc,test.auc=bk.test.nnet.auc)**

**##### [4] NAIVE BAYES #####**

**set.seed(1234)**

**bk.train.nb.fit <- naiveBayes(model,data=bk.train.df)**

**# area under ROC curve for TRAINING data**

**bk.train.df$nb.predprob <- as.numeric(predict(bk.train.nb.fit,**

**newdata=bk.train.df,**

**type="raw")[,2])**

**bk.train.nb.prediction <- prediction(bk.train.df$nb.predprob,bk.train.df$retain)**

**bk.train.nb.auc <- as.numeric(performance(bk.train.nb.prediction,"auc")@y.values)**

**# area under ROC curve for TEST data**

**bk.test.df$nb.predprob <- as.numeric(predict(bk.train.nb.fit,**

**newdata=bk.test.df,**

**type="raw")[,2])**

**bk.test.nb.prediction <- prediction(bk.test.df$nb.predprob,bk.test.df$retain)**

**bk.test.nb.auc <- as.numeric(performance(bk.test.nb.prediction,"auc")@y.values)**

**# ROC for naive Bayes classification**

**bk.train.nb.roc <- performance(bk.train.nb.prediction,"tpr","fpr")**

**bk.test.nb.roc <- performance(bk.test.nb.prediction,"tpr","fpr")**

**plot.roc(train.roc=bk.train.nb.roc,train.auc=bk.train.nb.auc,**

**test.roc=bk.test.nb.roc,test.auc=bk.test.nb.auc)**

**##### [5] RANDOM FORESTS #####**

**set.seed(1234)**

**bk.train.rf.fit <- randomForest(model,data=bk.train.df,mtry=3,**

**importance=FALSE,ntree=1000)**

**# area under ROC curve for TRAINING data**

**bk.train.df$rf.predprob <- as.numeric(predict(bk.train.rf.fit,type="prob")[,2])**

**bk.train.rf.prediction <- prediction(bk.train.df$rf.predprob,bk.train.df$retain)**

**bk.train.rf.auc <- as.numeric(performance(bk.train.rf.prediction,"auc")@y.values)**

**# area under ROC curve for TEST data**

**bk.test.df$rf.predprob <- as.numeric(predict(bk.train.rf.fit,newdata=bk.test.df,**

**type="prob")[,2])**

**bk.test.rf.prediction <- prediction(bk.test.df$rf.predprob,bk.test.df$retain)**

**bk.test.rf.auc <- as.numeric(performance(bk.test.rf.prediction,"auc")@y.values)**

**# ROC for random forest classification**

**bk.train.rf.roc <- performance(bk.train.rf.prediction,"tpr","fpr")**

**bk.test.rf.roc <- performance(bk.test.rf.prediction,"tpr","fpr")**

**plot.roc(train.roc=bk.train.rf.roc,train.auc=bk.train.rf.auc,**

**test.roc=bk.test.rf.roc,test.auc=bk.test.rf.auc)**

1. **Summary of Area Under Roc Curve**

# !! THIS PART HELPS WITH QUESTION 2 !!

**rbind(bk.train.lr.auc,**

**bk.train.svm.auc,**

**bk.train.nnet.auc,**

**bk.train.nb.auc,**

**bk.train.rf.auc)**

**rbind(bk.test.lr.auc,**

**bk.test.svm.auc,**

**bk.test.nnet.auc,**

**bk.test.nb.auc,**

**bk.test.rf.auc)**

1. **Prediction Of New Customer**

# !! THIS PART HELPS WITH QUESTION 3 !!

**bk\_new.df<- data.frame(age="15-24",income="50K-75K",tenure=0,district="Northeast Texas",**

**profit15=0,online15="No",billpay15="No")**

**bk\_new.df$nnet.predprob<- as.numeric(predict(bk.train.nnet.fit,newdata=bk\_new.df,type="raw"))**

**bk\_new.df$nnet.predYN<- predict(bk.train.nnet.fit,newdata=bk\_new.df,type="class")**