Binary Classification with Logistic Regression

Claire Boetticher, MSDS 410, SEC 56

November 8, 2020

Section 1: Introduction

The Universal Banking dataset contains information about candidates for personal loans, used for approval purposes. The objective of this analysis is to build a selection of logistic regression models for binary classification, specifically the development and evaluation of scoring classifiers with the goal of high discrimination ability between sample members' likelihood of opening a personal loan. These models are built in a predictive modeling framework, with an initial exploratory data analysis to examine incidence rates and potentially-useful predictor variables for modeling, cross-validation applied to assess models' performance in-sample and predictive accuracy out-of-sample, and statistical model evaluation using selected accuracy- and fit-related metrics.

Section 2: Sample Definition and Split

The original dataset includes 5000 observations of 14 variables. PersonalLoan is the dichotomous response variable, and all remaining 11 predictor variables are preserved except ZIP.Code and ID. All observations are kept in the sample for this analysis and no predictor variables are missing values.

Table 1: Universal Banking Variables

Variable
PersonalLoan (response)
Age
Experience
Income
Family
CCAvg
Education
Mortgage
SecuritiesAccount
CDAccount
Online
CreditCard

Section 1.2: Train/Test Split

In order to assess the logistic regression models' performance both in- and out-of-sample, basic cross-validation with a 70/30 train test split is applied to divide the sample. 70 percent of the 5000 observations are used for in-sample model development and 30 percent are used for out-of-sample model assessment. Table 2 shows resulting counts for subsequent model development and testing, reflecting the split.

Table 2: Train and Test Split

Dataset	Count	Percent of Total
Train	3492	69.84
Test	1508	30.16
Total	5000	100

Section 3: Exploratory Data Analysis

Traditional exploratory data analysis (EDA) for logistic regression is performed on the training dataset for examination of potential candidate predictor variables for modeling and to highlight notable trends. It is worth noting the imbalance in the response variable, PersonalLoan, with percentages shown for the full, train, and test datasets in Table 3.

Table 3: PersonalLoan Distribution by Response

	0	1
Original Dataset	0.904	0.096
Train	0.9015	0.0985
Test	0.9098	0.0902

Incidence rates for categorical predictor variables are calculated and plotted to display distribution of variables prior to modeling (Figure 1, light blue plots).

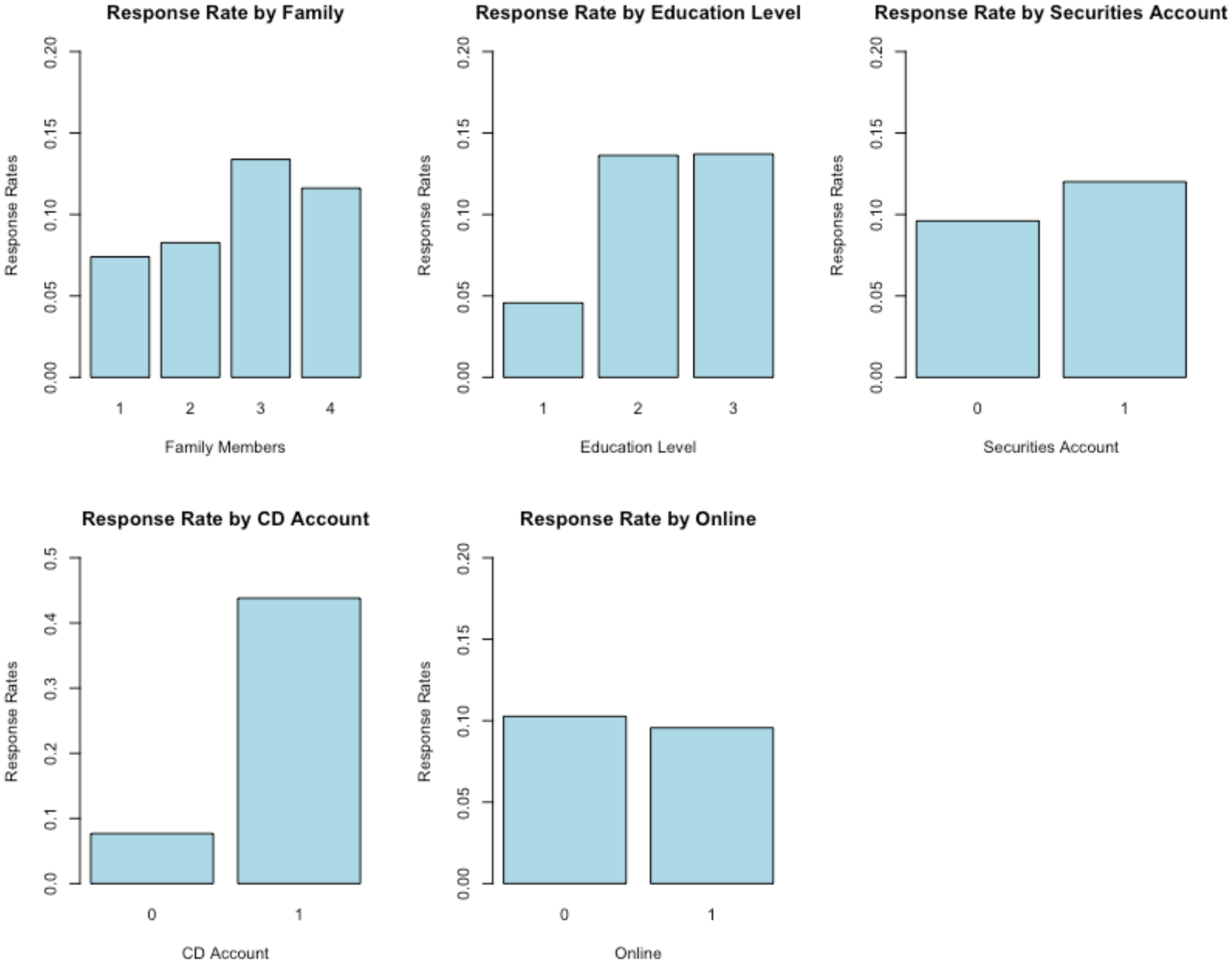

Figure 1: Incidence Rates for Categorical Predictor Variables

Response rate for PersonalLoan increases notably by Education Level from 1 to 2, with levels two and three comparably higher, suggesting a potential relationship worth consideration for modeling (Figure 1). An even larger increase is seen with CDAccount, with much higher incidence of personal loan applications for customers holding CD Accounts (Figure 1). Additionally, continuous predictor variables are discretized using the cut function, computing means for the response variable, PersonalLoan, in each cut (Figure 2, dark blue plots). The mean computations represent the probability that a personal loan is opened (i.e., that y = 1).

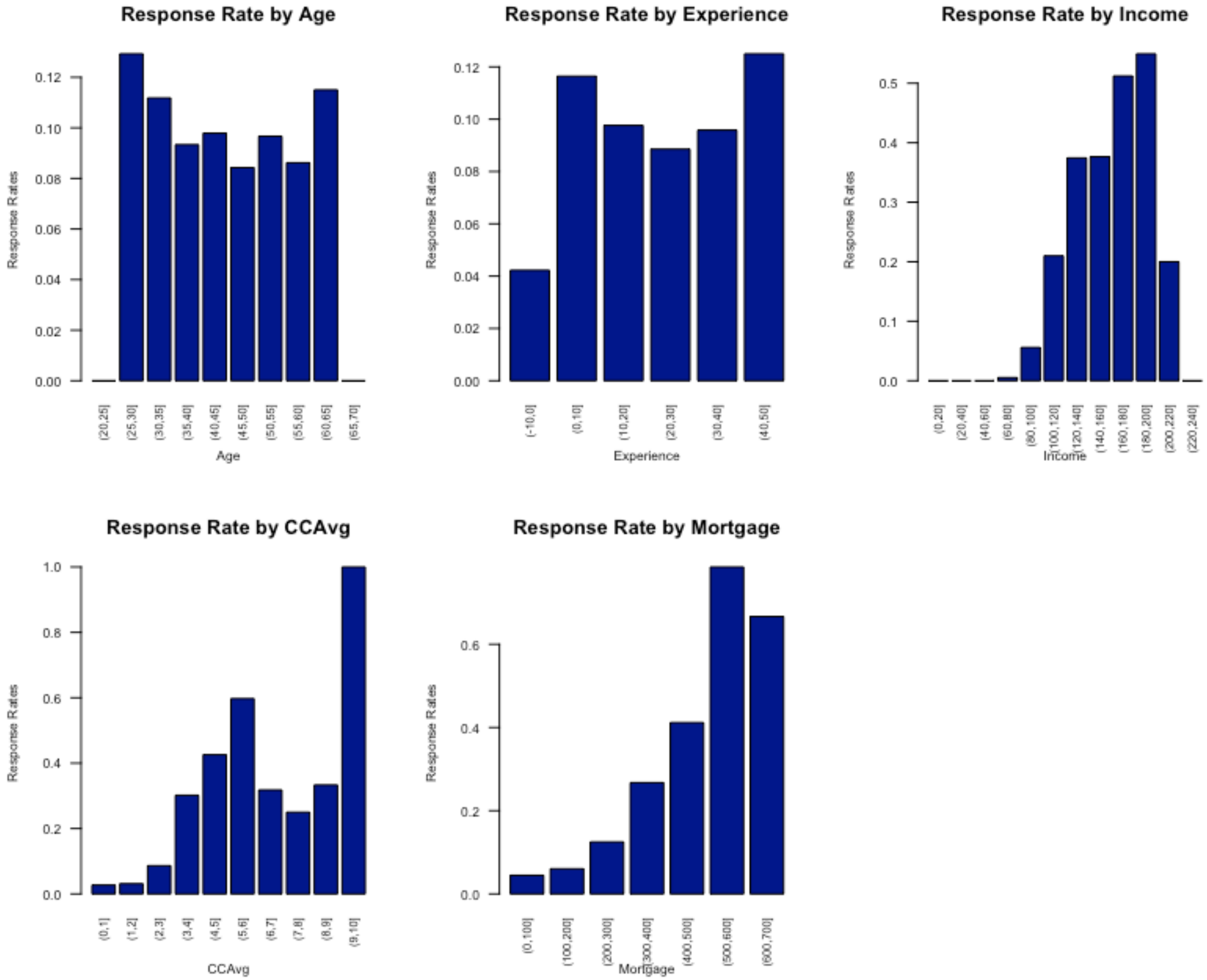

Figure 2: Incidence Rates for Discretized Predictor Variables

The variable Age shows larger response rates at the lower and higher end of the age spectrum for the sample, at the 25 to 35 range and again at the 60 and older range (Figure 2). Rates for Experience rise at the lower end of the spectrum, from -3 (possibly a mis-entry in the data) to 10.5, then levels off and remains roughly the same across the other cuts. Income shows a steady, sometimes-sharp increase in response rate, making this variable a viable candidate for modeling. Figure 3 shows the boxplot of Income distribution by Personal Loan responses, with a red cutoff line representing the 75th percentile value for response equaling one, supporting this possibility further.

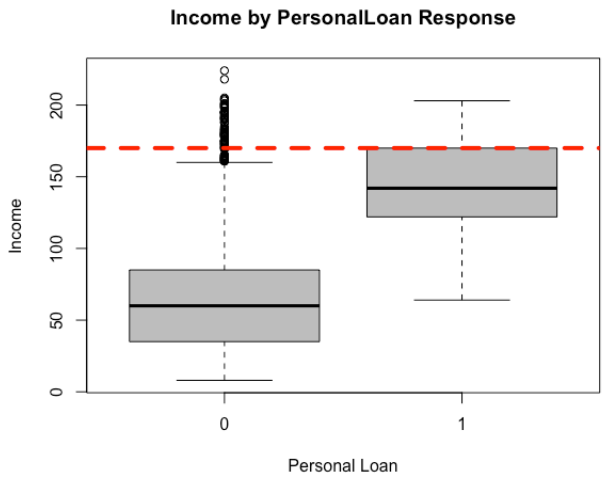

Figure 3: Income Distribution Boxplot

The plot for CCAvg suggests a trend in response rates, increasing up to values of six, then lowering steadily and rising sharply with values from 9 to 10. The boxplot in Figure 4 shows relatively clean separation between distributions, suggesting another predictor option for modeling.

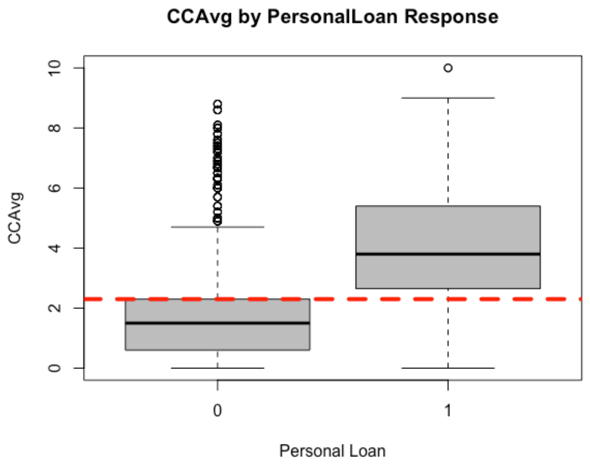

Figure 4: CCAvg Distribution Boxplot

Mortgage, once discretized, shows a steady increase in response rate as Mortgage values increase; however, the box plot for this variable shows highly-overlapping distributions, so that will be a factor in its inclusion in modeling design (Figure 5).

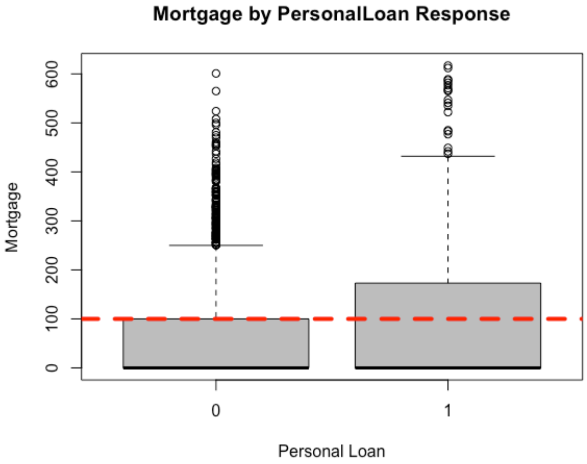

Figure 5: Mortgage Distribution Boxplot

No particularly noteworthy trends for this sample are seen with the predictors Online or Credit Card.

Section 4: Model Identification and In-Sample Model Fit

Three logistic regression models are identified and fit for evaluation of discrimination ability of scoring classifiers. These Generalized Linear Models (GLM) work by way of a link function, with parameter estimation using maximum likelihood estimation. The conditional mean of the response variable, PersonalLoan (y), is modeled through this link function. The Receiver Operating Characteristic (ROC) curve is produced to assist in the assessment of the discrimination of the fitted logistic models, plotted between the True Positive Rate (Sensitivity, representing the proportion of true positives correctly classified) along the Y axis and False Positive Rate (one minus Specificity, representing the proportion of true negatives correctly classified) along the X axis. The curve's appearance gives a visual perspective on discrimination ability, where a model with high ability with have both high sensitivity and specificity at the same time. The curve for a performant model with high discrimination ability would be close to the upper left corner of the plot; the 45-degree line on the plot represents a model with no discrimination ability, for comparison.

Additionally the Area under the Curve (AUC) metric, representing the performance of the ROC curve, is produced as a means for later model comparison. In theory, the higher the area (up to a maximum value of 1), the 'better' a model fits. Finally, a confusion matrix using the 'best' threshold value recommended by the curve is produced as a further point of evaluation and comparison for model performance in-sample, reflecting model sensitivity, specificity, and accuracy.

Section 4.1: Baseline Model

Six variables are selected for the baseline logistic regression model. Table 4 shows regression results for the baseline model for this analysis.

Table 4: Baseline Model

	Dependent variable:
	PersonalLoan
Income	0.06***
	(0.003)
CCAvg	0.16***
	(0.05)
CDAccount	2.49***
	(0.33)
factor(Education)2	4.08***
	(0.31)
factor(Education)3	4.10***
	(0.31)
Family	0.57***
	(0.09)
SecuritiesAccount	-0.57*
	(0.34)
Constant	-13.51***
	(0.66)
Observations	3,492
Log Likelihood	-439.54
Akaike Inf. Crit.	895.07
Note:	*p**p***p<0.01

All predictor variables except SecuritiesAccount are statistically significant; likewise, SecuritiesAccount is the only predictor with a negative relationship with PersonalLoan. Examining z-statistics for the coefficients, Income and Education (levels 2 and 3) have the highest values. All variables except SecuritiesAccount have z-statistic scores suggesting that there is statistical evidence that they are related to PersonalLoan, the response variable.

The ROC curve provides further visual basis for evaluating this model's discrimination ability (Figure 6), with a calculated AUC metric of 0.9584. The shape of the curve and the proximity of the AUC metric to 1 both suggest decent discrimination ability and fit, however this graph and the AUC value serve primarily as points of comparison for other models in the analysis, not as objective criteria for model performance and fit in and of themselves.

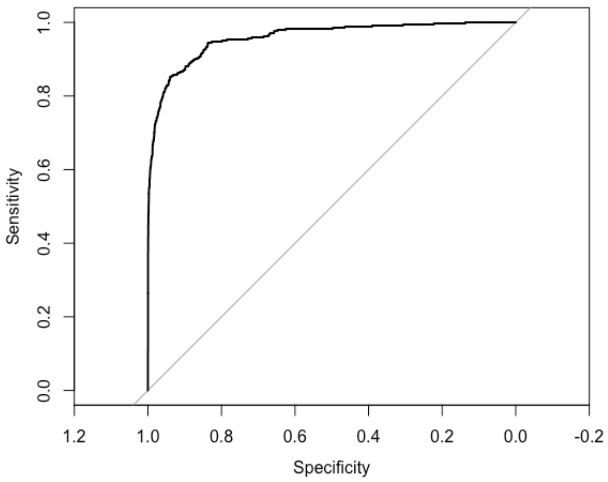

Figure 6: Baseline Model ROC Curve, AUC = 0.9584

Finally, a confusion matrix using the 'best' threshold value recommended by the curve (0.1657 in model 1) is produced as a further point of evaluation and comparison for model performance in-sample, reflecting model sensitivity, specificity, and accuracy.

Table 5: Baseline Model Confusion Matrix

	0 (Predicted)	1 (Predicted)
0 (Actual)	0.9400	0.0600
1 (Actual)	0.1483	0.8517

Sensitivity, or the True Positive Rate (TPR), at 85.17 percent, reflects decent classification discrimination between 0 and 1, no personal loan application versus an application. Specificity, or True Negative Rate (TNR), is higher at 94 percent. The overall accuracy, calculated as the sum of TPR and TNR divided by all observations, is 93.13 percent, also reflecting decent classification accuracy and discrimination. These will be compared with additional models to determine the comparative value of those metrics.

Section 4.2: Stepwise Variable Selection Model

The stepwise variable selection technique begins with a simple linear regression to initialize stepwise selection, in this case with Income as the predictor variable. Income is selected because of its appearance of a strong relationship with the response variable PersonalLoan and also the visual separation of distributions by response variable (Figure 2). At each stage through selection, variables are included into or deleted from the model based on the criteria of Akaike's information criterion (AIC) for the stepwise procedure, where addition is terminated once the addition of a variable causes no reduction in AIC. AIC is essentially a metric enabling assessment of model fit and simplicity. Table 6 shows regression results for the model with this technique applied.

Table 6: Stepwise Variable Selection

	Dependent variable:
	PersonalLoan
Income	0.06***
	(0.003)
Education	1.75***
	(0.13)
CDAccount	3.38***
	(0.38)
Family	0.66***
	(0.09)
Online	-0.73***
	(0.18)
CreditCard	-0.79***
	(0.23)
SecuritiesAccount	-0.90***
	(0.34)
CCAvg	0.12***
	(0.05)
Experience	0.01*
	(0.01)
Constant	-13.55***
	(0.71)
Observations	3,492
Log Likelihood	-473.11
Akaike Inf. Crit.	966.21
Note:	*p**p***p<0.01

Compared to the baseline model, the stepwise model selected all included plus Experience, Online, and CreditCard for inclusion based on AIC contribution. All predictor variables except Experience are statistically significant. The z-statistics support this, with all variables except Experience with scores suggesting that there is statistical evidence that they are related to PersonalLoan, the response variable. The AIC, at 966.21, is notably higher than the baseline model.

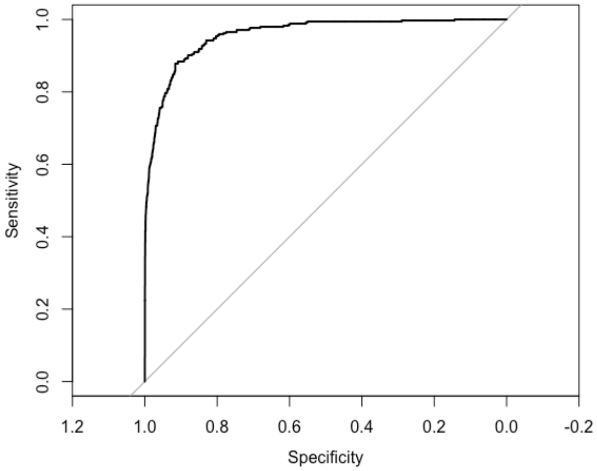

Figure 7: Stepwise Model ROC Curve, AUC = 0.9578

The ROC curve provides further visual basis for evaluating this model's discrimination ability (Figure 7), with a calculated AUC metric of 0.9578. The shape of the curve and the proximity of the AUC metric to 1 both suggest decent discrimination ability and fit.

The confusion matrix using the 'best' threshold value recommended by the curve (0.1404 in the stepwise model) results are provided in Table 7.

Table 7: Stepwise Model Confusion Matrix

	0 (Predicted)	1 (Predicted)
0 (Actual)	0.9158	0.0842
1 (Actual)	0.1221	0.8779

Sensitivity, or the True Positive Rate, at 87.79 percent, reflects decent classification discrimination between 0 and 1, no personal loan application versus an application. Specificity, or True Negative Rate, is higher at 91.58 percent. The overall accuracy, calculated as the sum of TPR and TNR divided by all observations, is 91.21 percent, also reflecting decent classification accuracy and discrimination.

Section 4.3: Simplified Model

A third model is built for comparison to the baseline model (six predictor variables mixed across financial measures (e.g., Income and CCAvg) and demographic measures (e.g., Family)) with the stepwise model initiated at Income. This simplified model uses four predictor variables – Income, CCAvg, Education (as a factor) and Family. It is a less complex model than the baseline and prioritizes candidate variables that show promising comparative distributions with the response variable (Figures 3 and 4) and merited inclusion in the stepwise model early on in that AIC-driven selection process.

Table 8: Simplified Model

	Dependent variable:
	PersonalLoan
Income	0.06***
	(0.003)
CCAvg	0.18***
	(0.05)
factor(Education)2	4.10***
	(0.30)
factor(Education)3	4.13***
	(0.30)
Family	0.56***
	(0.08)
Constant	-13.39***
	(0.63)
Observations	3,492
Log Likelihood	-471.14
Akaike Inf. Crit.	954.27
Note:	*p**p***p<0.01

All model coefficients from regression results are statistically significant (Table 8). Examining the z-statistics, all variables have values suggesting that there is statistical evidence that they are related to PersonalLoan, the response variable. The AIC value is higher for the Stepwise model than model 1, though, indicating a possibly-inferior fit and suggesting that the two predictors missing from this model but included in the baseline model – SecuritiesAccount and CDAccount – could contribute meaningfully to discrimination ability.

The ROC curve provides further visual basis for evaluating this model's discrimination ability (Figure 8), with a calculated AUC metric of 0.9501, only slightly smaller than model 1. The shape of the curve toward the upper left and the proximity of the AUC metric to 1 both suggest decent discrimination ability and fit.

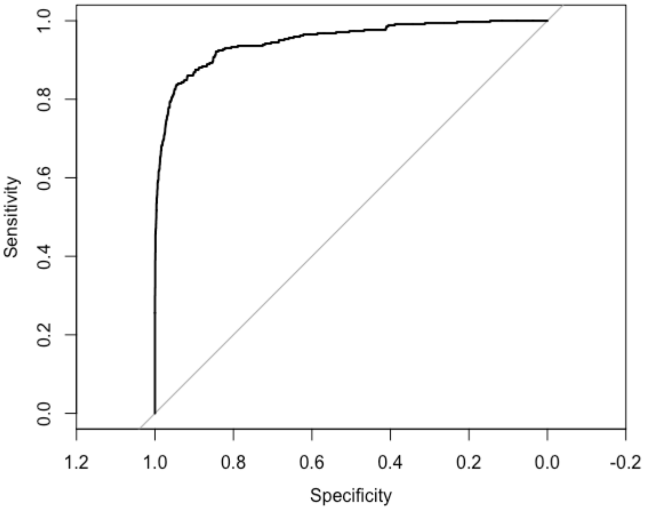

Figure 8: Simplified Model ROC Curve, AUC = 0.9501

The confusion matrix, using the 'best' threshold value recommended by the curve (0.1932 in model 3), is shown in Table 9.

Table 9: Simplified Model Confusion Matrix

	0 (Predicted)	1 (Predicted)
0 (Actual)	0.9450	0.0550
1 (Actual)	0.1628	0.8372

Sensitivity, or the True Positive Rate, at 83.72 percent, reflects decent classification discrimination between 0 and 1, no personal loan application versus an application. Specificity, or True Negative Rate, is higher at 94.50 percent. The overall accuracy, calculated as the sum of TPR and TNR divided by all observations, is 93.44 percent, also reflecting decent classification accuracy and discrimination.

Section 4.4: Model Comparison

From a visual perspective, the ROC curve graphics do not provide a meaningful-enough difference to claim any potential difference in fit, though they all take the shape of a curve that could possibly represent a model that discriminates decently. The quantitative metrics calculated in Table 9 show further metrics calculated across all three models as a more reliable means of assessing performance in-sample. AIC is added as well, a measure for assessing model fit and complexity based on deviance (and designed to prevent irrelevant predictors' inclusion). It is not interpretable on its own but provides a means for model comparison. Table 10 shows comparative results.

Table 10: Model Predictive Accuracy Metrics and Ranking (In-Sample)

Model	AUC (Rank)	Accuracy (Rank)	AIC (Rank)
Baseline	0.9584 (1)	0.9313 (2)	895.07 (1)
Stepwise	0.9578 (2)	0.9121 (3)	966.21 (3)
Simplified	0.9501 (3)	0.9344 (1)	954.27 (2)

In-sample, AUC measures for all three models are quite close in value, as are accuracy measures. The baseline model out-performs the stepwise and simplified models in terms of AUC and AIC, both proxies for model fit. The difference in AIC measures from the baseline model (ranking first) and the other two is more defined. These results in-sample suggest that at least for the training data, the baseline model's predictor variables, together, capture the relationship best with the response variable and provide a more reliable means for discriminating as a scoring classifier for this sample. Though the third model is less complex in theory and has slightly higher accuracy and fewer predictors included in the baseline, these results suggest that the added predictors could be, in fact, contributing to discrimination ability and model fit. Further exploration is warranted before leaving out any baseline predictor variables.

Section 5: Predictive Accuracy

To assess how well each model performs out-of-sample, thus its potential utility for predictive modeling in future classification tasks, accuracy metrics are calculated using the test dataset. Table 11 results show True Positive Rate and accuracy calculations for each model.

Table 11: Model Predictive Accuracy Metrics and Ranking (Out-of-Sample)

Model	TPR (Rank)	Accuracy (Rank)
Baseline	0.77059 (1)	0.9629 (1)
Stepwise	0.7059 (2)	0.9595 (2)
Simplified	0.6765 (3)	0.9595 (2)

All three models' classifying performance was lower out-of-sample, not too surprising on unseen data. The baseline model outperforms the stepwise and simplified models in both TPR and accuracy metrics, by a slightly higher margin with TPR; accuracy for all three models are very similar. Depending on the threshold for 'success' in correctly classifying true positives, in hopes of leading to a customer likely to pursue a personal loan, the Stepwise and Simplified models may warrant further investigation and development given their respective lower TPR scores (even though all three models have relatively high and comparable accuracy scores). Ranking of TPR out-of-sample – with Baseline, Stepwise, then Simplified performance in that order – mirrors that of AUC in-sample. The imbalance in the response variable may lead to a decision to a rank ordering of scoring classifier effectiveness in this case, as opposed to a defined threshold. Given the overlap in the predictor variables included across all three models, further investigation of potential relationships amongst predictors would potentially shed light on the extent of their contribution to discrimination ability. Ultimately, the potential outcome and criticality (financial/business-wise) of classifying a potential customers' likelihood of applying for a personal loan versus not should drive whether these results are sufficient as outputs for decision-making processes.

R script

```
# Assignment_8.R

# Claire Boetticher

# 11.8.2020


# Load dependencies
library(pROC)
library(stargazer)
library(MASS)
library(car)

######################################################################
# Load and inspect data
######################################################################

# Set working directory
path <- "/Users/clb/Documents/MSDS410/8-Logistic_Regression/"
setwd(path)

# Read in csv file for banking data;
path.name <- '/Users/clb/Documents/MSDS410/8-Logistic_Regression/data/';
file.name <- paste(path.name,'UniversalBank.csv',sep='');
my.data <- read.csv(file.name,header=TRUE);

str(my.data)
head(my.data)

# Drop ZIP.code and ID from dataset
my.data <- subset(my.data,select=-c(ZIP.Code,ID))

######################################################################
# Section 2: Train Test split
######################################################################
# Set the seed on the random number generator to get the same split every time
code is run
# 0-1: 70% interval becomes 70% training set
set.seed(12345)
my.data$u <- runif(n=dim(my.data)[1],min=0,max=1)

# Create train/test split
train.df <- subset(my.data, u<.70)
test.df <- subset(my.data, u>=.70)

# Check data split. Sum of parts should equal whole
# 5000 is sample size
# 3492 = ~70% of sample
# 1508 = ~30% of sample
dim(my.data)[1]
dim(train.df)[1]
dim(test.df)[1]
dim(train.df)[1]+dim(test.df)[1]
3492/5000
1508/5000

# Check for missing values
```

```
sapply(train.df, function(x) sum(is.na(x)))

# Drop u column from train and test df
train.df <- subset(train.df,select=-c(u))
test.df <- subset(test.df,select=-c(u))


#####################################################################
# Section 3: Exploratory Data Analysis
#####################################################################

# Incidence rates with barplots

# Selected boxplots = the standard EDA plot for discrete predictor variables

# We are looking at the degree of separation, can one be superimposed?
# Boxplot is IQR: if separate, max classification error is 25% in a single
variable
# Completely separated means log reg doesn't matter, though classification is
perfect

#####################################################################
# Section 3.1: Response rates for discrete variables
#####################################################################

par(mfrow=c(1,1))

# PersonalLoan distribution
table(my.data$PersonalLoan)
table(train.df$PersonalLoan)
table(test.df$PersonalLoan)

# Family
response.Family <- aggregate(train.df$PersonalLoan,
                                by=list(Family=train.df$Family),
                                FUN=mean
)

barplot(height=response.Family$x,names.arg=response.Family$Family,
        xlab='Family Members',ylab='Response Rate',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Number of Family Members')

# Boxplot - Family ~ PersonalLoan - not good separation in 2-3 member group
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$Family,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))

# predictor ~ response, with x response, y predictor
boxplot(Family ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='Family', col='grey')
abline(h=3,col='red',lwd=4,lty=2)
title('Family Members by PersonalLoan Response')

# Education
response.Education <- aggregate(train.df$PersonalLoan,
           by=list(Education=train.df$Education),
           FUN=mean
```

```
)

barplot(height=response.Education$x,names.arg=response.Education$Education,
      xlab='Education Level',ylab='Response Rate',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Education Level')

# Boxplot - Education ~ PersonalLoan - not good separation
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$Education,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))

# predictor ~ response, with x response, y predictor
boxplot(Education ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='Education', col='grey')
abline(h=3,col='red',lwd=4,lty=2)
title('Education Level by PersonalLoan Response')

# SecuritiesAccount
response.SecuritiesAccount <- aggregate(train.df$PersonalLoan,

by=list(SecuritiesAccount=train.df$SecuritiesAccount),
                                FUN=mean
)

barplot(height=response.SecuritiesAccount$x,names.arg=response.SecuritiesAccount$
SecuritiesAccount,
        xlab='Securities Account',ylab='Response
Rate',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Securities Account')

# CDAccount
response.CDAccount <- aggregate(train.df$PersonalLoan,
                                        by=list(CDAccount=train.df$CDAccount),
                                        FUN=mean
)

barplot(height=response.CDAccount$x,names.arg=response.CDAccount$CDAccount,
        xlab='CD Account',ylab='Response Rate',ylim=c(0,0.5),col='lightblue')
title('Response Rate by CD Account')

# Online
response.Online <- aggregate(train.df$PersonalLoan,
                                by=list(Online=train.df$Online),
                                FUN=mean
)

barplot(height=response.Online$x,names.arg=response.Online$Online,
        xlab='Online',ylab='Response Rate',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Online')

# CreditCard
response.CreditCard <- aggregate(train.df$PersonalLoan,
                             by=list(Online=train.df$CreditCard),
                             FUN=mean
)

barplot(height=response.CreditCard$x,names.arg=response.CreditCard$CreditCard,
        xlab='Credit Card',ylab='Response Rate',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Credit Card')



#####################################################################
# Section 3.2: Discretize continuous variables
#####################################################################

help(cut)
# When breaks is specified as a single number, the range of the data is divided
into breaks pieces of
# equal length, and then the outer limits are moved away by 0.1% of the range to
ensure that the extreme
# values both fall within the break intervals.

# Age
# Initial distribution
summary(train.df$Age)

# Discretize variable into bins
age.cuts <- cut(x=train.df$Age,breaks=seq(20,80,5))
# age.cuts <- cut(x=train.df$Age,breaks=10)
levels(age.cuts)
table(age.cuts)

age.personalloan <- aggregate(train.df$PersonalLoan,by=list(age.cuts=age.cuts),
                            FUN=mean)

# Barplot
barplot(height=age.personalloan$x, names.arg=age.personalloan$age.cuts,
        las=2,col='darkblue',xlab='Age',ylab='Response Rates',cex.axis =
.8,cex.names = .8)
title('Response Rate by Age')

# Age ~ PersonalLoan - not good separation
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$Age,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))

# predictor ~ response, with x response, y predictor
boxplot(Age ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='Age', col='grey')
abline(h=54.25,col='red',lwd=4,lty=2)
title('Age by PersonalLoan Response')

# Experience
# Initial distribution
summary(train.df$Experience)
# table(train.df$Experience)
# barplot(height=train.df$Experience, xlab='Experience', ylab='Response Rate')

# Discretize variable into bins
experience.cuts <- cut(x=train.df$Experience,breaks=seq(-10,50,10))
# experience.cuts <- cut(x=train.df$Experience,breaks=10)
levels(experience.cuts)
table(experience.cuts)
```

```
experience.personalloan <-
aggregate(train.df$PersonalLoan,by=list(experience.cuts=experience.cuts),
                              FUN=mean)

# Barplot
barplot(height=experience.personalloan$x,
names.arg=experience.personalloan$experience.cuts,
        las=2,col='darkblue',xlab='Experience',ylab='Response Rates',cex.axis =
.8,cex.names = .8)
title('Response Rate by Experience')

# Experience ~ PersonalLoan - not good separation
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$Experience,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))

# predictor ~ response, with x response, y predictor
boxplot(Experience ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='Experience', col='grey')
abline(h=29.25,col='red',lwd=4,lty=2)
title('Experience by PersonalLoan Response')

# Income
# Initial distribution
summary(train.df$Income)
# barplot(height=train.df$Income, xlab='Income', ylab='Response Rate')

# Discretize variable into bins - need to mess with bins
income.cuts <- cut(x=train.df$Income,breaks=seq(0,240,20))
# income.cuts <- cut(x=train.df$Income,breaks=10)
levels(income.cuts)
table(income.cuts)

income.personalloan <-
aggregate(train.df$PersonalLoan,by=list(income.cuts=income.cuts),
                                     FUN=mean)

# Barplot
barplot(height=income.personalloan$x, names.arg=income.personalloan$income.cuts,
        las=2,col='darkblue',xlab='Income',ylab='Response Rates',cex.axis =
.8,cex.names = .8)
title('Response Rate by Income')

# Income ~ PersonalLoan - great separation - show!
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$Income,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))

# predictor ~ response, with x response, y predictor
boxplot(Income ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='Income', col='grey')
abline(h=170,col='red',lwd=4,lty=2)
title('Income by PersonalLoan Response')

# CCAvg
# Initial distribution
```

```
summary(train.df$CCAvg)
# barplot(height=train.df$CCAvg, xlab='CCAvg', ylab='Response Rate')

# Discretize variable into bins - need to mess with bins
# ccavg.cuts <- cut(x=train.df$CCAvg,breaks=seq(0,10,1))
ccavg.cuts <- cut(x=train.df$CCAvg,breaks=10)
levels(ccavg.cuts)
table(ccavg.cuts)

ccavg.personalloan <-
aggregate(train.df$PersonalLoan,by=list(ccavg.cuts=ccavg.cuts),
                                 FUN=mean)

# Barplot
barplot(height=ccavg.personalloan$x, names.arg=ccavg.personalloan$ccavg.cuts,
        las=2,col='darkblue',xlab='CCAvg',ylab='Response Rates',cex.axis =
.8,cex.names = .8)
title('Response Rate by CCAvg')

# CCAvg ~ PersonalLoan - great separation - show!
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$CCAvg,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))

# predictor ~ response, with x response, y predictor
boxplot(CCAvg ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='CCAvg', col='grey')
abline(h=2.3,col='red',lwd=4,lty=2)
title('CCAvg by PersonalLoan Response')

# Mortgage
# Initial distribution
summary(train.df$Mortgage)
# barplot(height=train.df$Mortgage, xlab='Mortgage', ylab='Response Rate')

# Discretize variable into bins - need to mess with bins
mortgage.cuts <- cut(x=train.df$Mortgage,breaks=seq(0,700,100))
# mortgage.cuts <- cut(x=train.df$Mortgage,breaks=10)
levels(mortgage.cuts)
table(mortgage.cuts)

mortgage.personalloan <-
aggregate(train.df$PersonalLoan,by=list(mortgage.cuts=mortgage.cuts),
                                FUN=mean)

# Barplot
barplot(height=mortgage.personalloan$x,
names.arg=mortgage.personalloan$mortgage.cuts,
        las=2,col='darkblue',xlab='Mortgage',ylab='Response Rates',cex.axis =
.8,cex.names = .8)
title('Response Rate by Mortgage')

# Mortgage ~ PersonalLoan - not good separation
# Use aggregate() function for abline, 75th percentile of 1 incidence
aggregate(train.df$Mortgage,by=list(PersonalLoan=train.df$PersonalLoan),
          FUN=quantile,p=c(0,0.25,0.5,0.75,1))
```

```
# predictor ~ response, with x response, y predictor
boxplot(Mortgage ~ train.df$PersonalLoan, data=train.df, xlab='Personal Loan',
        ylab='Mortgage', col='grey')
abline(h=100,col='red',lwd=4,lty=2)
title('Mortgage by PersonalLoan Response')

######################
# CCAvg Bins - example
######################

# my.data$CCAvg_Bins <- cut(my.data$CCAvg,breaks=20)
# table(my.data$CCAvg_Bins)
#
# response.CCAvg_Bins <- aggregate(my.data$PersonalLoan,
#          by=list(CCAvg_Bins=my.data$CCAvg_Bins),
#          FUN=mean
# );
#
# barplot(height=response.CCAvg_Bins$x,names.arg=response.CCAvg_Bins$CCAvg_Bins,
#    xlab='CCAvg_Bin',ylab='Response Rate',las=2,cex.names=0.75)

######################################################################
# Panel plots for report
######################################################################

# Incidence rates (rows by columns)
par(mfrow=c(2,3))

barplot(height=response.Family$x,names.arg=response.Family$Family,
        xlab='Family Members',ylab='Response
Rates',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Family')

barplot(height=response.Education$x,names.arg=response.Education$Education,
        xlab='Education Level',ylab='Response
Rates',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Education Level')

barplot(height=response.SecuritiesAccount$x,names.arg=response.SecuritiesAccount$
SecuritiesAccount,
        xlab='Securities Account',ylab='Response
Rates',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Securities Account')

barplot(height=response.CDAccount$x,names.arg=response.CDAccount$CDAccount,
        xlab='CD Account',ylab='Response Rates',ylim=c(0,0.5),col='lightblue')
title('Response Rate by CD Account')

barplot(height=response.Online$x,names.arg=response.Online$Online,
        xlab='Online',ylab='Response Rates',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Online')

barplot(height=response.CreditCard$x,names.arg=response.CreditCard$CreditCard,
        xlab='Credit Card',ylab='Response Rate',ylim=c(0,0.2),col='lightblue')
title('Response Rate by Credit Card')

# Incidence rates (rows by columns)
```

```
par(mfrow=c(2,3))

barplot(height=age.personalloan$x, names.arg=age.personalloan$age.cuts,
        las=2,col='darkblue',xlab='Age',ylab='Response Rates',cex.axis =
0.8,cex.names = .7,cex.lab = .8)
title('Response Rate by Age')

barplot(height=experience.personalloan$x,
names.arg=experience.personalloan$experience.cuts,
        las=2,col='darkblue',xlab='Experience',ylab='Response Rates',cex.axis =
0.8,cex.names = .7,cex.lab = .8)
title('Response Rate by Experience')

barplot(height=income.personalloan$x, names.arg=income.personalloan$income.cuts,
        las=2,col='darkblue',xlab='Income',ylab='Response Rates',cex.axis =
0.8,cex.names = .7,cex.lab = .8)
title('Response Rate by Income')

barplot(height=ccavg.personalloan$x, names.arg=ccavg.personalloan$ccavg.cuts,
        las=2,col='darkblue',xlab='CCAvg',ylab='Response Rates',cex.axis =
0.8,cex.names = .7,cex.lab = .8)
title('Response Rate by CCAvg')

barplot(height=mortgage.personalloan$x,
names.arg=mortgage.personalloan$mortgage.cuts,
        las=2,col='darkblue',xlab='Mortgage',ylab='Response Rates',cex.axis =
0.8,cex.names = .7,cex.lab = .8)
title('Response Rate by Mortgage')


#####################################################################
# Section 4: Fit a Naive Model
#####################################################################

# What happens if I forget to specify the family argument?
model.1a <- glm(PersonalLoan ~ Income+CCAvg+CDAccount+factor(Education)+Family
           +SecuritiesAccount, data=train.df)
summary(model.1a)

model.1b <- lm(PersonalLoan ~ Income+CCAvg+CDAccount+factor(Education)+Family
           +SecuritiesAccount, data=train.df, family=('binomial'))
summary(model.1b)


#####################################################################
# Section 4.1: Fit a Naive Model as baseline
#####################################################################

help(glm)
# Fit a basic logistic regression model;
# Use this as a scoring classifier, some # between zero and 1
# Higher scores closer to response = 1, binary classification

model.1 <- glm(PersonalLoan ~ Income+CCAvg+CDAccount+factor(Education)+Family
               +SecuritiesAccount, data=train.df, family=('binomial'))

summary(model.1)
```

```
# Model output table
out.path <- '/Users/clb/Documents/MSDS410/8-Logistic_Regression/report_outputs/'
file.name <- 'model1.html';
stargazer(model.1, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Baseline Model'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

# Does this model fit well?  How should we evaluate this model?
# This is a binary classification model - a scoring classifier.
# We need to evaluate the classification accuracy.

#####################################################################
# ROC curve classification
#####################################################################

# Model object
names(model.1)

# Create vector of model scores
model.score.1 <- model.1$fitted.values
summary(model.score.1)

#######################################################################
# Generate ROC curve and plot it
#######################################################################
# Note that we are using model scores to generate the ROC curve

roc.1 <- roc(response=train.df$PersonalLoan, predictor=model.1$fitted.values)
print(roc.1)
plot(roc.1)

# Compute AUC
auc.1 <- auc(roc.1)

#> auc.1
#Area under the curve: 0.9584

#######################################################################
# Find the threshold value recommended by the ROC curve
#######################################################################

roc.specs <- coords(roc=roc.1,x=c('best'),
input=c('threshold','specificity','sensitivity'),
ret=c('threshold','specificity','sensitivity'),
as.list=TRUE
)

#######################################################################
# Once we have the threshold value assign classes
#######################################################################

train.df$model1scores <- model.1$fitted.values
train.df$model1classes <- ifelse(train.df$model1scores>roc.specs$threshold,1,0)

# Rough confusion matrix using counts
table(train.df$PersonalLoan, train.df$model1classes)
```

```
# Proper confusion matrix
t <- table(train.df$PersonalLoan, train.df$model1classes)

# Compute row totals
r <- apply(t,MARGIN=1,FUN=sum)

# Normalize confusion matrix to rates
t/r

# Accuracy calculation:
# (TP+TN)/(TP+TN+FP+FN)
(293+2959)/(2959+189+51+293)

# Look at your confusion matrix and compare it to roc.specs.
# Do we see anything interesting?
# What values are on the diagonal? specificity (TN,upper left) and sensitivity
(TP,lower right)
# Specificity (FPR): 0.93996188
# Sensitivity (TPR): 0.85174419
# What values are on the off-diagonal? FP(upper right) and FN(lower left)

# Create score dataframe and remove unneeded columns from train.df
score.df <- data.frame(train.df$model1scores,train.df$model1classes)
train.df <- subset(train.df,select=-c(model1classes,model1scores))

#####################################################################
# Section 4.2: Stepwise Variable Selection Model with stepAIC()
#####################################################################

# Fit a basic logistic regression model;
# Use this as a scoring classifier, some # between zero and 1
# Higher scores closer to response = 1, binary classification

# Define upper and lowers models for stepAIC()
# AIC is penalized likelihood approach to variable selection, smallest AIC wins
# Emphasizes trade-off between model fit and complexity

help(stepAIC)

# Define the upper model as the FULL model
upper.glm <- glm(PersonalLoan ~ .,data=train.df,family=('binomial'))
summary(upper.glm)

# Define the lower model as the Intercept model
lower.glm <- glm(PersonalLoan ~ 1,data=train.df,family=('binomial'))
summary(lower.glm)

# Need a SLR to initialize stepwise selection, Income selected as best for now
given separation from boxplot (??)
income.glm <- glm(PersonalLoan ~ Income,data=train.df,family=('binomial'))
summary(income.glm)

# StepAIC()
stepwise.glm <-
stepAIC(object=income.glm,scope=list(upper=formula(upper.glm),lower=~1),
                      direction=c('both'))
```

```
summary(stepwise.glm)
vif(stepwise.glm)

# extractAIC function
# Returns a numeric vector of length 2, with first and second elements giving
edf,
# the ‘equivalent degrees of freedom’ for the fitted model fit + AIC, the
(generalized) Akaike
# Information Criterion for fit
extractAIC(stepwise.glm)

# Model output table
out.path <- '/Users/clb/Documents/MSDS410/8-Logistic_Regression/report_outputs/'
file.name <- 'stepwiseglm.html';
stargazer(stepwise.glm, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX: Stepwise Variable Selection'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

#####################################################################
# ROC curve classification
#####################################################################

# Model object
names(stepwise.glm)

# Create vector of model scores
model.score.stepwise <- stepwise.glm$fitted.values
summary(model.score.stepwise )

#######################################################################
# Generate ROC curve and plot it
#######################################################################
# Note that we are using model scores to generate the ROC curve

roc.2 <- roc(response=train.df$PersonalLoan,
predictor=stepwise.glm$fitted.values)
print(roc.2)
plot(roc.2)

# Compute AUC
auc.2 <- auc(roc.2)

#> auc.2
#Area under the curve:

#######################################################################
# Find the threshold value recommended by the ROC curve
#######################################################################

roc.specs <- coords(roc=roc.2,x=c('best'),
                    input=c('threshold','specificity','sensitivity'),
                    ret=c('threshold','specificity','sensitivity'),
                    as.list=TRUE
)


#######################################################################
```

```
# Once we have the threshold value assign classes
#######################################################################

train.df$model2scores <- stepwise.glm$fitted.values
train.df$model2classes <- ifelse(train.df$model2scores>roc.specs$threshold,1,0)

# Rough confusion matrix using counts
table(train.df$PersonalLoan, train.df$model2classes)

# Proper confusion matrix
t <- table(train.df$PersonalLoan, train.df$model2classes);

# Compute row totals
r <- apply(t,MARGIN=1,FUN=sum)

# Normalize confusion matrix to rates
t/r

# Accuracy calculation
# (TP+TN)/(TP+TN+FP+FN)
(302+2883)/(2883+265+42+302)

# Look at your confusion matrix and compare it to roc.specs.
# Do we see anything interesting?
# What values are on the diagonal? specificity (TN,upper left) and sensitivity
(TP,lower right)
# What values are on the off-diagonal? FP(upper right) and FN(lower left)

# Add to score dataframe and remove unneeded columns from train.df
score.df$model2scores <- train.df$model2scores
score.df$model2classes <- train.df$model2classes
train.df <- subset(train.df,select=-c(model2classes,model2scores))

#####################################################################
# Section 4.3: Simplified MODEL
#####################################################################

model.3 <- glm(PersonalLoan ~ Income+CCAvg+factor(Education)+Family,
               data=train.df, family=('binomial'))

summary(model.3)

# Model output table
out.path <- '/Users/clb/Documents/MSDS410/8-Logistic_Regression/report_outputs/'
file.name <- 'model3.html';
stargazer(model.3, type=c('html'),out=paste(out.path,file.name,sep=''),
          title=c('Table XX:Simplified Model'),
          align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

# Does this model fit well?  How should we evaluate this model?
# This is a binary classification model - a scoring classifier.
# We need to evaluate the classification accuracy.

#####################################################################
# ROC curve classification
#####################################################################
```

```
# Model object
names(model.3)

# Create vector of model scores
model.score.3 <- model.3$fitted.values
summary(model.score.3)

#######################################################################
# Generate ROC curve and plot it
#######################################################################
# Note that we are using model scores to generate the ROC curve

roc.3 <- roc(response=train.df$PersonalLoan, predictor=model.3$fitted.values)
print(roc.3)
plot(roc.3)

# Compute AUC
auc.3 <- auc(roc.3)

#> auc.1
#Area under the curve: 0.9584

#######################################################################
# Find the threshold value recommended by the ROC curve
#######################################################################

roc.specs <- coords(roc=roc.3,x=c('best'),
                    input=c('threshold','specificity','sensitivity'),
                    ret=c('threshold','specificity','sensitivity'),
                    as.list=TRUE
)

#######################################################################
# Once we have the threshold value assign classes
#######################################################################

train.df$model3scores <- model.3$fitted.values
train.df$model3classes <- ifelse(train.df$model3scores>roc.specs$threshold,1,0)

# Rough confusion matrix using counts
table(train.df$PersonalLoan, train.df$model3classes)

# Proper confusion matrix
t <- table(train.df$PersonalLoan, train.df$model3classes)

# Compute row totals
r <- apply(t,MARGIN=1,FUN=sum)

# Normalize confusion matrix to rates
t/r

# Accuracy calculation:
# (TP+TN)/(TP+TN+FP+FN)
(288+2975)/(2975+173+56+288)

# Look at your confusion matrix and compare it to roc.specs.
# Do we see anything interesting?
# What values are on the diagonal? specificity (TN,upper left) and sensitivity
(TP,lower right)
# What values are on the off-diagonal? FP(upper right) and FN(lower left)

# Add to score dataframe and remove unneeded columns from train.df
score.df$model3scores <- train.df$model3scores
score.df$model3classes <- train.df$model3classes
train.df <- subset(train.df,select=-c(model3classes,model3scores))

#####################################################################
# Section 5: Predictive Accuracy
#####################################################################
# Evaluate out of sample performance for all 3 models

#####################################################################
# Section 5.1: Predictive Accuracy for Model 1
#####################################################################
model1.test <- predict(model.1,newdata=test.df,type="response")
model1.test <- ifelse(model1.test > 0.5,1,0)
test.df$model1.test <- model1.test

# Rough confusion matrix using counts
table(test.df$PersonalLoan, test.df$model1.test)

# Proper confusion matrix
t <- table(test.df$PersonalLoan, test.df$model1.test)

# Compute row totals
r <- apply(t,MARGIN=1,FUN=sum)

# Normalize confusion matrix to rates
t/r

# Accuracy calculation
# (TP+TN)/(TP+TN+FP+FN)
(96+1356)/(1356+16+40+96)


#####################################################################
# Section 5.2: Predictive Accuracy for Stepwise Model
#####################################################################
model2.test <- predict(stepwise.glm,newdata=test.df,type="response")
model2.test <- ifelse(model2.test > 0.5,1,0)
test.df$model2.test <- model2.test

# Rough confusion matrix using counts
table(test.df$PersonalLoan, test.df$model2.test)

# Proper confusion matrix
t <- table(test.df$PersonalLoan, test.df$model1.test)

# Compute row totals
r <- apply(t,MARGIN=1,FUN=sum)

# Normalize confusion matrix to rates
t/r
```

```
# Accuracy calculation
# (TP+TN)/(TP+TN+FP+FN)
(92+1355)/(1355+17+44+92)

#####################################################################
# Section 5.3: Predictive Accuracy for Simplified Model
#####################################################################
model3.test <- predict(model.3,newdata=test.df,type="response")
model3.test <- ifelse(model3.test > 0.5,1,0)
test.df$model3.test <- model3.test

# Rough confusion matrix using counts
table(test.df$PersonalLoan, test.df$model3.test)

# Proper confusion matrix
t <- table(test.df$PersonalLoan, test.df$model3.test)

# Compute row totals
r <- apply(t,MARGIN=1,FUN=sum)

# Normalize confusion matrix to rates
t/r

# Accuracy calculation
# (TP+TN)/(TP+TN+FP+FN)
(92+1355)/(1355+17+44+92)
```