# MSDS 410

# Assignment 8

## Section 1: Data Split and EDA

### 1.1: Data Split

This assignment's goal is to build logistic regression models to classify and predict if a person is likely to open a personal loan. The data consists of 5000 observations with 13 predictor variables and one binary response variable, PersonalLoan, which indicates whether a person has a personal loan (1) or does not (0). The data is split into a training set and test set at a roughly 70%/30% ratio. The specific counts are shown in Figure 1.

| Data Split | Observation Count |
|---|---|
| Training Data | 3492 |
| Testing Data | 1508 |

*Figure 1: Observation Counts*

### 1.2: EDA

Once the dataset has been split, EDA is performed by computing the incidence rates for important categorical and continuous variables. The bar graphs representing these rates for education, family, CCAvg, age, experience, and income are shown below in Figure 2. Bins have been created for continuous variables to discretize them, making it easy to visualize in a bar chart. The graphs show that response rates follow different patterns depending on the variable used to compute them. Some predictors, such as age and experience, follow an approximate uniform distribution with low variance between bins. Other predictors like income and education level show a clear increase in response rate as the factor level increases.

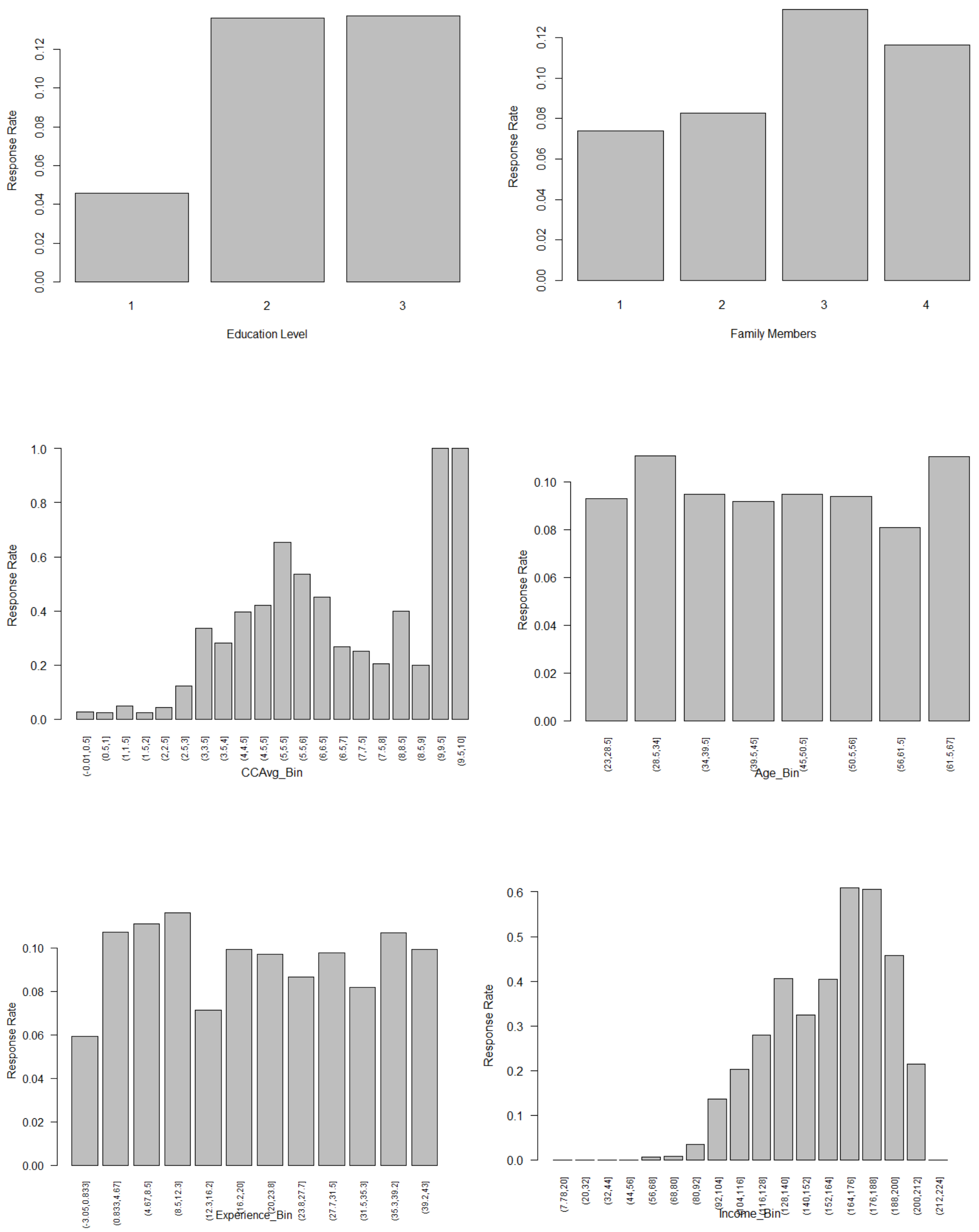


*Figure 2: Bar Graphs of Incidence Rates for Different Variables*

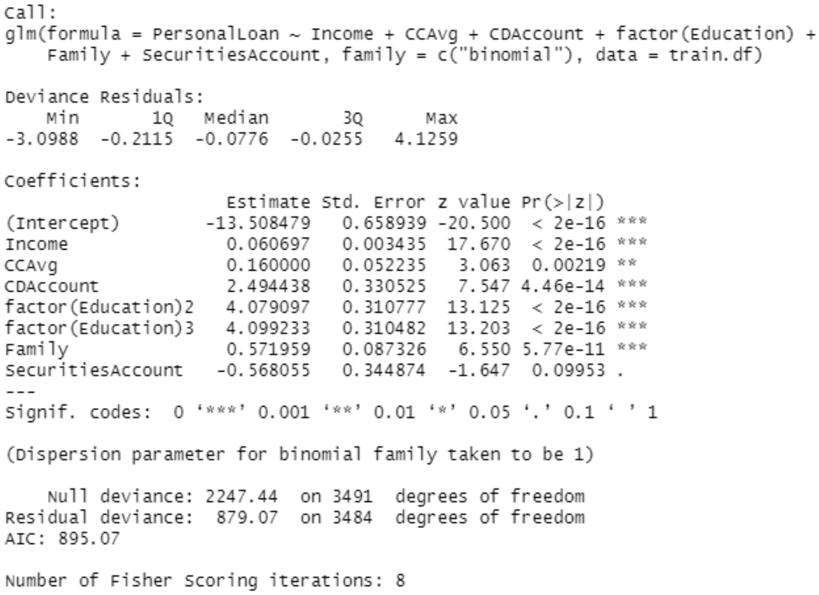
## Section 2: Modeling

### 2.1: Naïve Model

A baseline Naïve model is created using the following formula:

PersonalLoan ~ Income+CCAvg+CDAccount+factor(Education)+Family+SecuritiesAccount

This model assumes a binomial distribution since we are predicting a yes/no for whether a person has a personal loan.



```
Call:
glm(formula = PersonalLoan ~ Income + CCAvg + CDAccount + factor(Education) +
    Family + SecuritiesAccount, family = c("binomial"), data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0988  -0.2115  -0.0776  -0.0255   4.1259

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -13.508479   0.658939 -20.500  < 2e-16 ***
Income              0.060697   0.003435  17.670  < 2e-16 ***
CCAvg               0.160000   0.052235   3.063  0.00219 **
CDAccount           2.494438   0.330525   7.547 4.46e-14 ***
factor(Education)2  4.079097   0.310777  13.125  < 2e-16 ***
factor(Education)3  4.099233   0.310482  13.203  < 2e-16 ***
Family              0.571959   0.087326   6.550 5.77e-11 ***
SecuritiesAccount  -0.568055   0.344874  -1.647  0.09953 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2247.44  on 3491  degrees of freedom
Residual deviance:  879.07  on 3484  degrees of freedom
AIC: 895.07

Number of Fisher Scoring iterations: 8
```

*Figure 3: Naive Model Summary*

The ROC graph for this model is shown below. Area under the curve is 0.9584

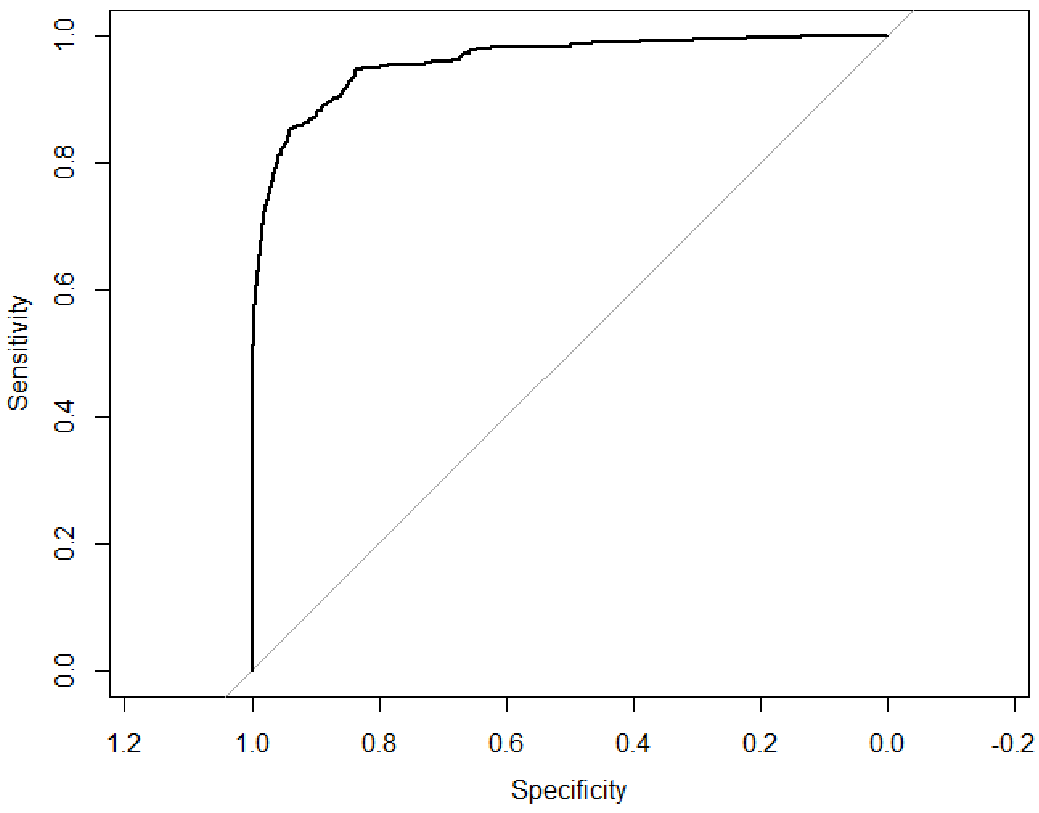
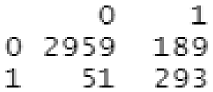


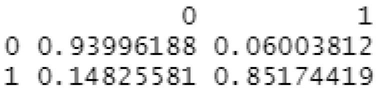
*Figure 4: ROC Curve for Model 1*

Next, a confusion matrix using both raw totals and percentages of correctly labeled and mislabeled observations is created. This model was able to correctly identify 94% of people without a personal loan and 85% of people with a personal loan.



```
     0    1
0 2959  189
1   51  293
```

*Figure 5: Confusion Matrix (Model 1 counts)*



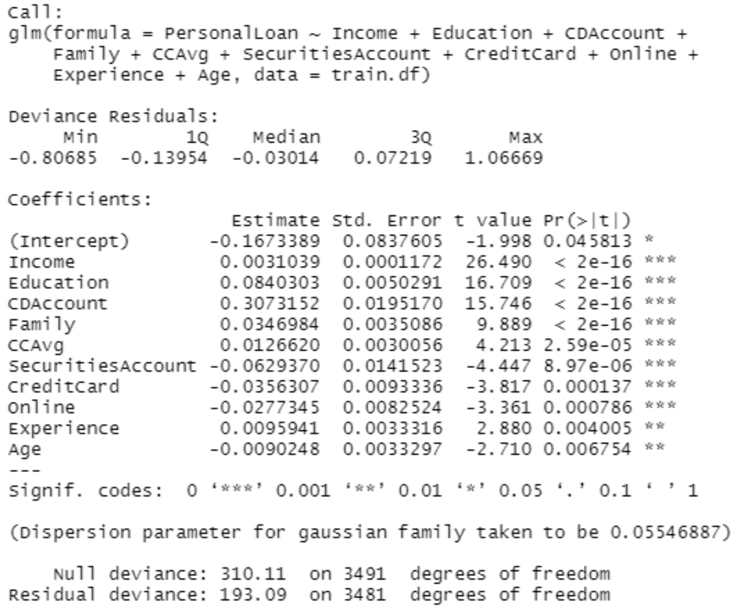
```
           0          1
0 0.93996188 0.06003812
1 0.14825581 0.85174419
```

*Figure 6: Confusion Matrix (Model 1 percent)*

### 2.2: StepAIC Model

Our second model is created by using forward variable selection using the full model at the upper bound and the intercept model as the lower bound. Using this method, the formula for the model was found to be:

PersonalLoan ~ Income + Education + CDAccount + Family + CCAvg + SecuritiesAccount + CreditCard + Online + Experience + Age



```
Call:
glm(formula = PersonalLoan ~ Income + Education + CDAccount + 
    Family + CCAvg + SecuritiesAccount + CreditCard + Online + 
    Experience + Age, data = train.df)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-0.80685  -0.13954  -0.03014   0.07219   1.06669  

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -0.1673389  0.0837605  -1.998 0.045813 *  
Income             0.0031039  0.0001172  26.490  < 2e-16 ***
Education          0.0840303  0.0050291  16.709  < 2e-16 ***
CDAccount          0.3073152  0.0195170  15.746  < 2e-16 ***
Family             0.0346984  0.0035086   9.889  < 2e-16 ***
CCAvg              0.0126620  0.0030056   4.213 2.59e-05 ***
SecuritiesAccount -0.0629370  0.0141523  -4.447 8.97e-06 ***
CreditCard        -0.0356307  0.0093336  -3.817 0.000137 ***
Online            -0.0277345  0.0082524  -3.361 0.000786 ***
Experience         0.0095941  0.0033316   2.880 0.004005 ** 
Age               -0.0090248  0.0033297  -2.710 0.006754 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.05546887)

    Null deviance: 310.11  on 3491  degrees of freedom
Residual deviance: 193.09  on 3481  degrees of freedom
```

*Figure 7: StepAIC Model Summary*

The ROC curve for this model is shown below in Figure 8. The area under the curve is 0.9577, very similar to our previous model despite the difference in predictor variables.

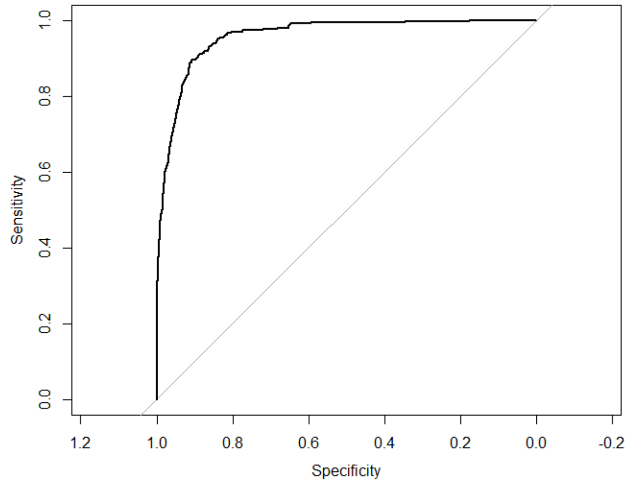
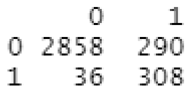


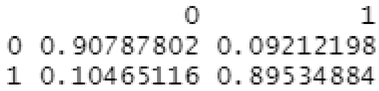
*Figure 8: ROC Curve (Model 2)*

Next, a confusion matrix using both raw totals and percentages of correctly labeled and mislabeled observations is created. This model was able to correctly identify 90% of people without a personal loan and 89% of people with a personal loan.



```
     0    1
0 2858  290
1   36  308
```

*Figure 9: Confusion Matrix (Model 2 counts)*



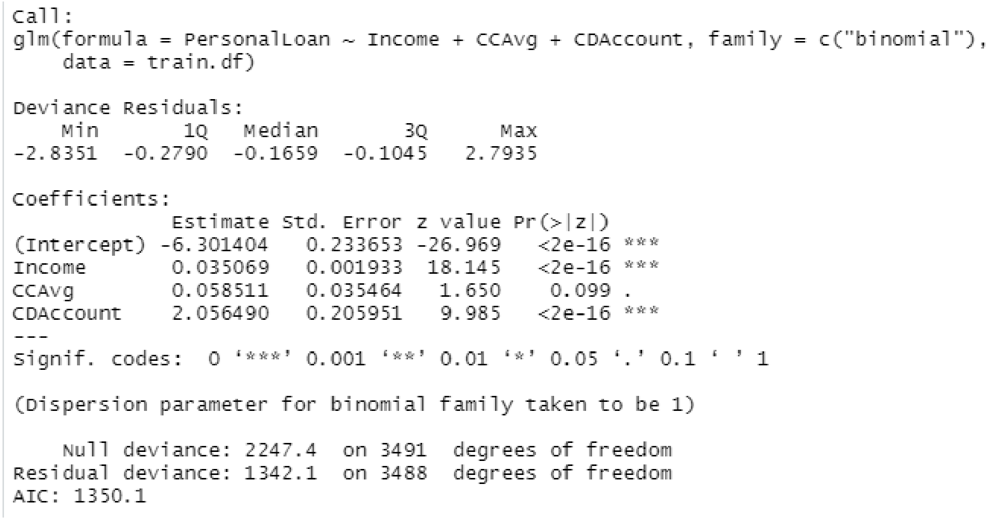
```
           0          1
0 0.90787802 0.09212198
1 0.10465116 0.89534884
```

*Figure 10: Confusion Matrix (Model 2 percent)*

## 2.3: Correlation Model

For the final model, variables were chosen based on their correlation to the response variable, PersonalLoan. None of the predictor variables had a very strong correlation (>0.5), so variables were chosen if their correlation coefficient was greater than 0.25. This resulted in the selection of three predictor variables, for a final model formula of:

PersonalLoan ~ Income+CCAvg+CDAccount



```
Call:
glm(formula = PersonalLoan ~ Income + CCAvg + CDAccount, family = c("binomial"),
    data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8351  -0.2790  -0.1659  -0.1045   2.7935

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.301404   0.233653 -26.969   <2e-16 ***
Income       0.035069   0.001933  18.145   <2e-16 ***
CCAvg        0.058511   0.035464   1.650    0.099 .
CDAccount    2.056490   0.205951   9.985   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2247.4  on 3491  degrees of freedom
Residual deviance: 1342.1  on 3488  degrees of freedom
AIC: 1350.1
```

*Figure 11: Summary of Correlation Model*

The ROC for model 3 is shown below. Area under the curve was calculated as 0.9315

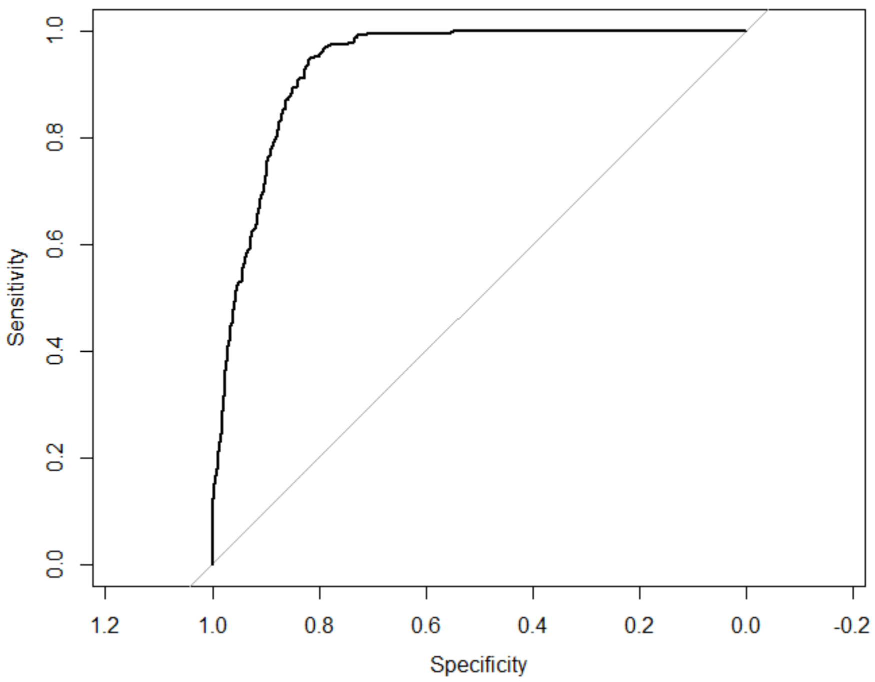
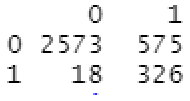


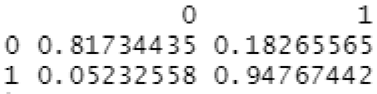
*Figure 12: ROC Curve (Model 3)*

Next, a confusion matrix using both raw totals and percentages of correctly labeled and mislabeled observations is created. This model was able to correctly identify 82% of people without a personal loan and 95% of people with a personal loan.



```
     0    1
0 2573  575
1   18  326
```

*Figure 13: Confusion Matrix (Model 3 counts)*



```
           0          1
0 0.81734435 0.18265565
1 0.05232558 0.94767442
```

*Figure 14: Confusion Matrix (Model 3 percent)*

## Section 3: Model Comparison and Predictive Accuracy

### 3.1: Model Comparison

Table 1 below shows a comparison of the 3 models that were built with respect to different calculated metrics.

| Model Number | AUC | % True Positive (Personal Loan) | % True Negative (No Personal Loan) |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | 0.9584 | 85 | 94 |
| 2 | 0.9577 | 89 | 90 |
| 3 | 0.9315 | 95 | 82 |

*Table 1: Metrics (Train Data)*

These results don't point to a clear 'best' model out of the three, but each model's effectiveness depends on the business problem at hand. The first model was the best at classifying observations with no personal loan, however it had the lowest percentage of correct personal loan identifications. The third model had opposite results; it was very good at classifying people with personal loans, but also generated a lot of false positives classifying observations without personal loans as people with one. The second model was somewhere in the middle of the first two, with good (but not great) rates of classification.

### 3.2: Predictive Accuracy (Test Data)

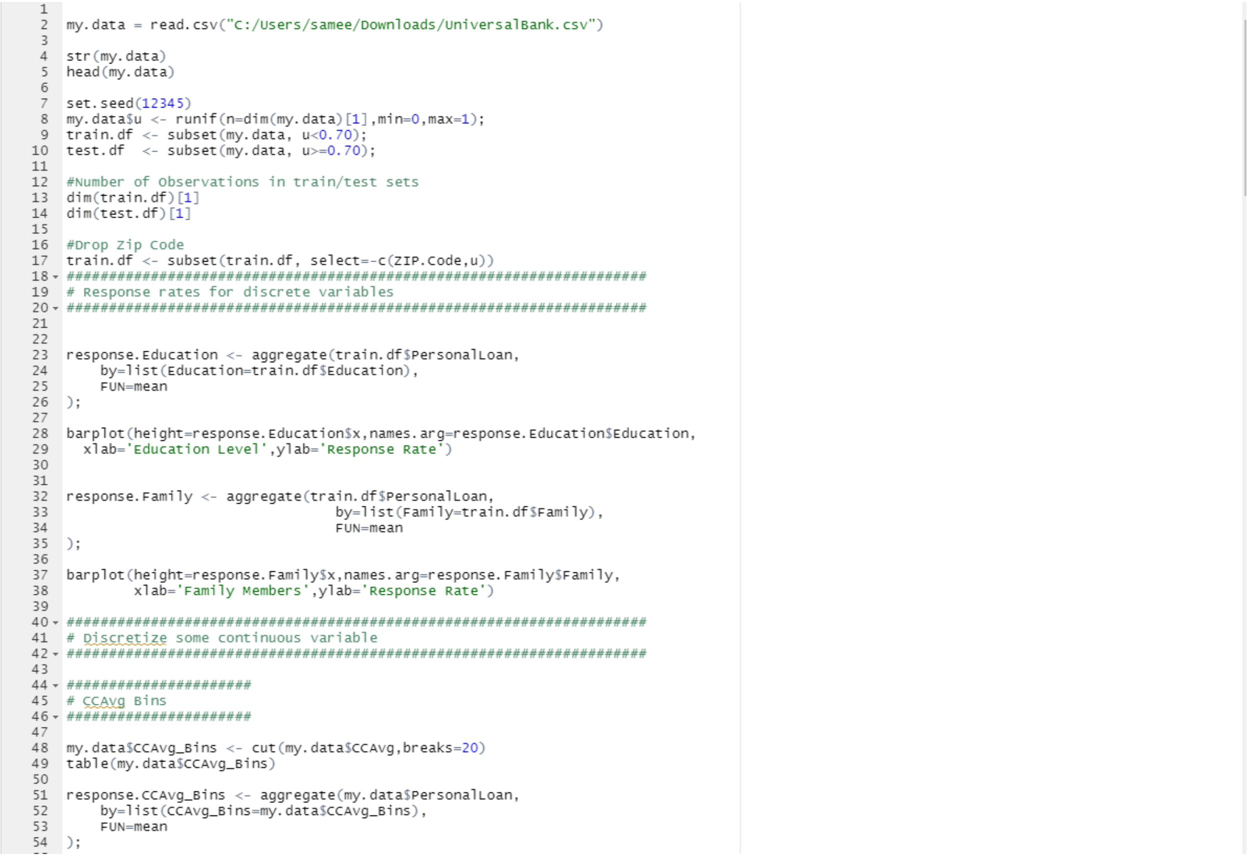
The same metrics are calculated for each model, but this time using the test data. This should get rid of any overfitting that may have occurred when using the training data and give us a more accurate view of how well each of the models can classify personal loans.

| Model Number | AUC | % True Positive (Personal Loan) | % True Negative (No Personal Loan) |
|---|---|---|---|
| 1 | 0.9591 | 91 | 86 |
| 2 | 0.9599 | 98 | 53 |
| 3 | 0.9389 | 96 | 82 |

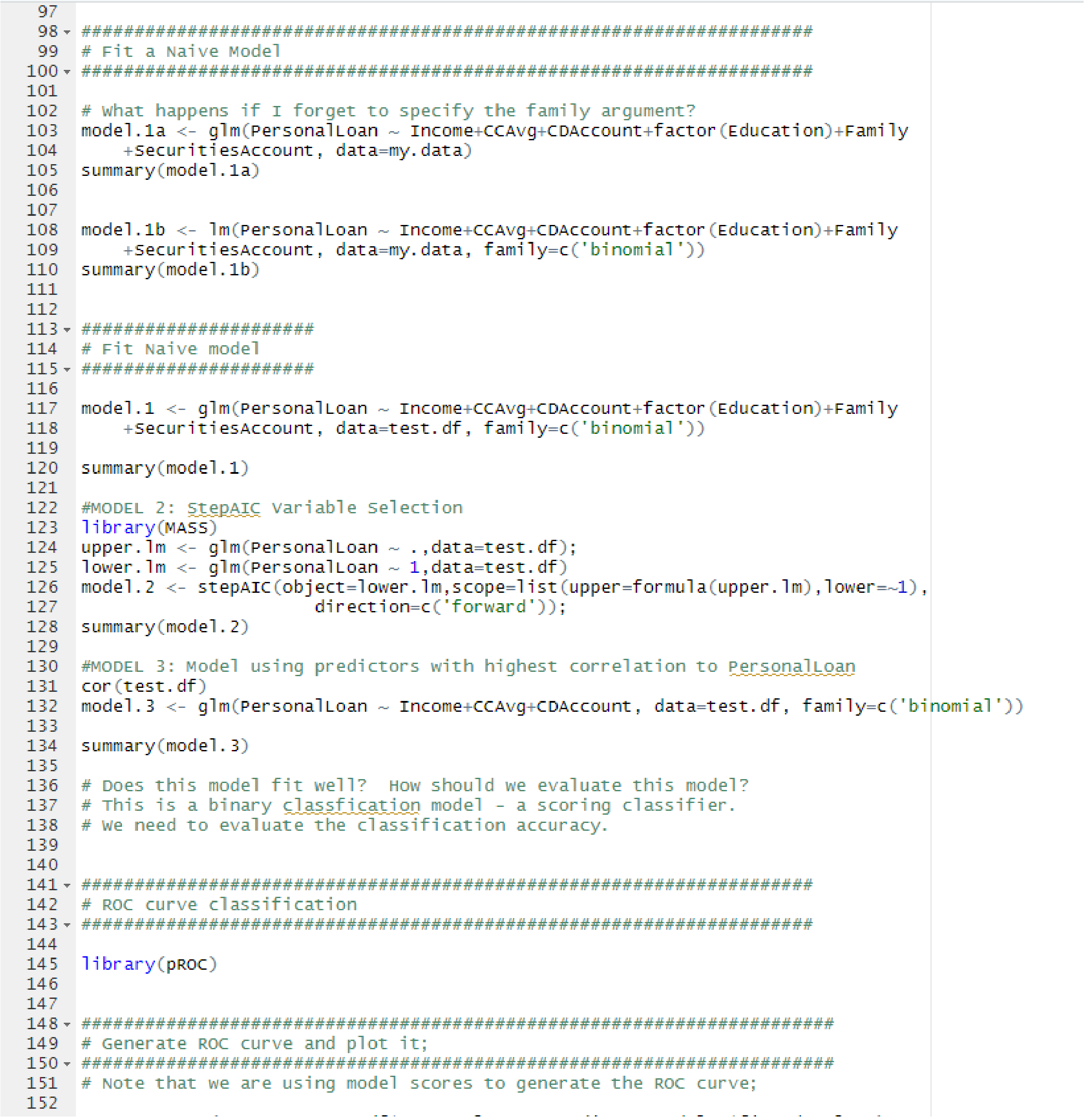
*Table 2: Metrics (Test Data)*

Our out-of-sample results were slightly different than our in-sample ones. The most change occurred with model 2 (stepAIC). In-sample, this model was the most balanced of the three, but using the test data we see that it tends to overclassify people as having personal loans. This causes the model to correctly identify 98% of personal loan cases but also incorrectly classifies almost half of the observations without personal loans. The first model saw a slight increase in the true positive rate, but a similar decrease in true negative rate. The third model had almost the same results. It is still unclear which model is “best”, but based on the test data results, models 1 or 3 seem to be the most consistent.

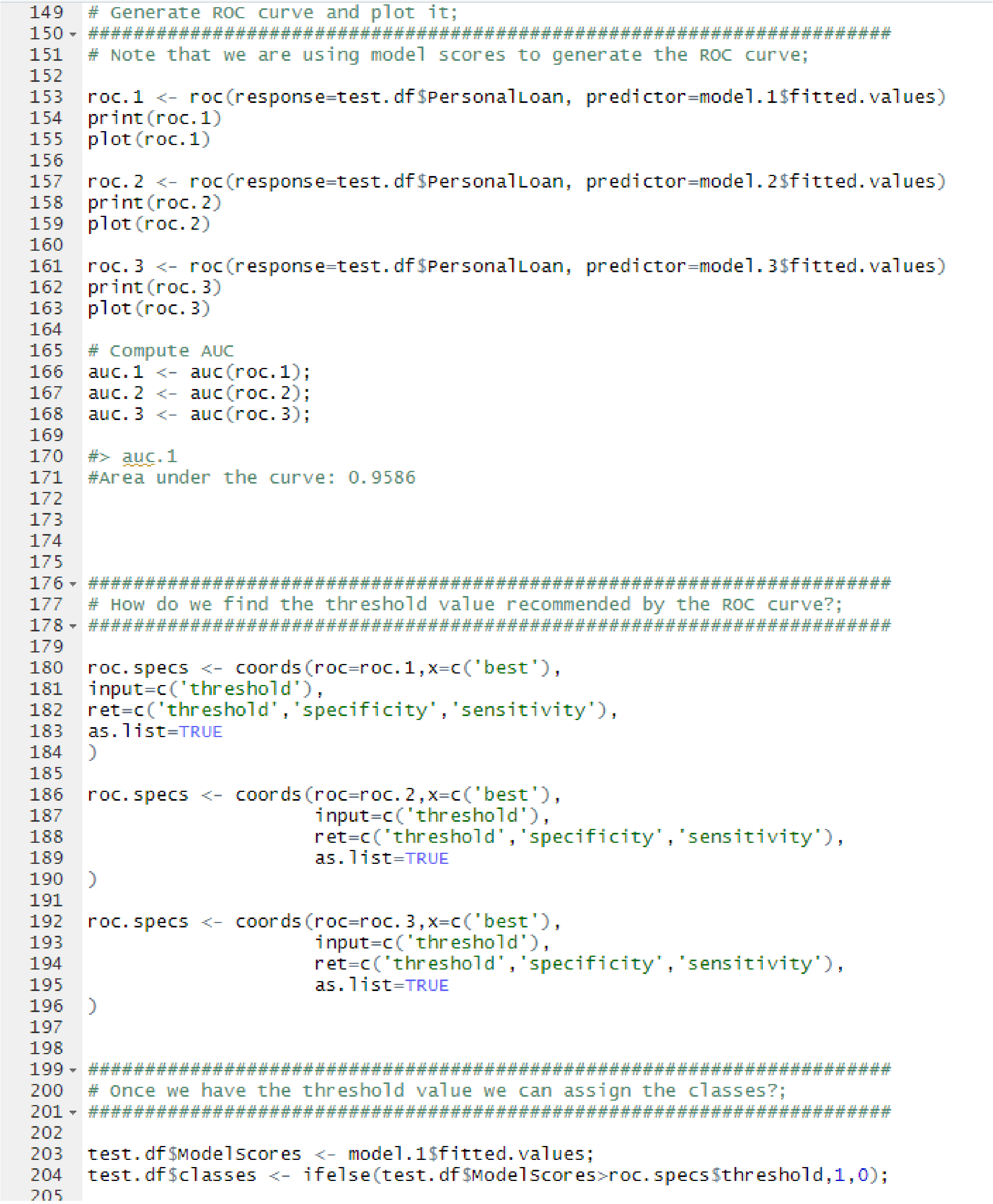
## Appendix



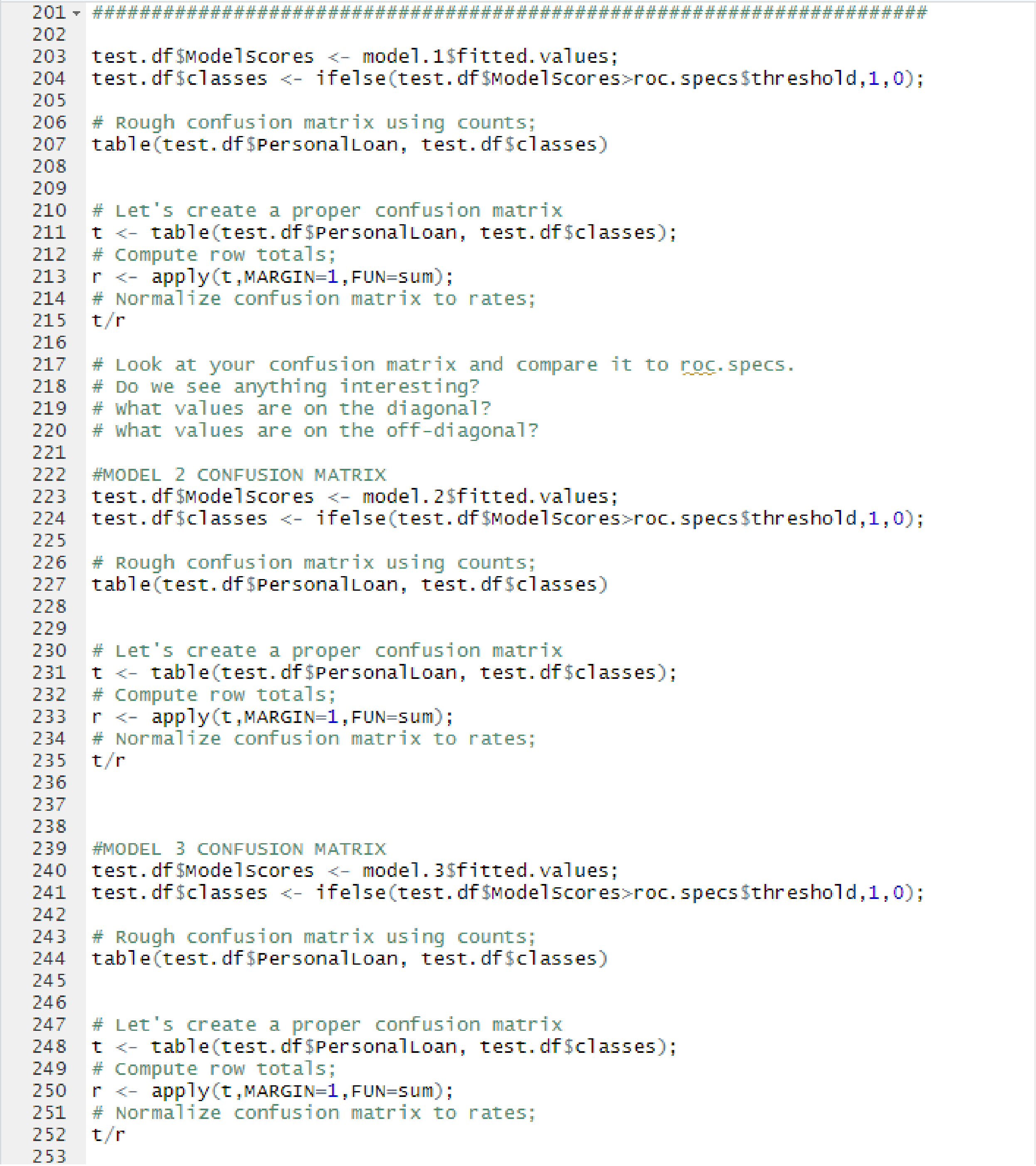
```r

my.data = read.csv("C:/Users/samee/Downloads/UniversalBank.csv")

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

set.seed(12345)
my.data$u <- runif(n=dim(my.data)[1],min=0,max=1);
train.df <- subset(my.data, u<0.70);
test.df  <- subset(my.data, u>=0.70);

#Number of Observations in train/test sets
dim(train.df)[1]
dim(test.df)[1]

#Drop Zip Code
train.df <- subset(train.df, select=-c(ZIP.Code,u))
#####################################################################
# Response rates for discrete variables
#####################################################################


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')


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')

#####################################################################
# Discretize some continuous variable
#####################################################################

######################
# CCAvg Bins
######################

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
);
```



```r

######################################################################
# 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=my.data)
summary(model.1a)


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


######################
# Fit Naive model
######################

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

summary(model.1)

#MODEL 2: StepAIC Variable Selection
library(MASS)
upper.lm <- glm(PersonalLoan ~ .,data=test.df);
lower.lm <- glm(PersonalLoan ~ 1,data=test.df)
model.2 <- stepAIC(object=lower.lm,scope=list(upper=formula(upper.lm),lower=~1),
                      direction=c('forward'));
summary(model.2)

#MODEL 3: Model using predictors with highest correlation to PersonalLoan
cor(test.df)
model.3 <- glm(PersonalLoan ~ Income+CCAvg+CDAccount, data=test.df, family=c('binomial'))

summary(model.3)

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


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

library(pROC)


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

```



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

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

roc.2 <- roc(response=test.df$PersonalLoan, predictor=model.2$fitted.values)
print(roc.2)
plot(roc.2)

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

# Compute AUC
auc.1 <- auc(roc.1);
auc.2 <- auc(roc.2);
auc.3 <- auc(roc.3);

#> auc.1
#Area under the curve: 0.9586




######################################################################
# How do we find the threshold value recommended by the ROC curve?;
######################################################################

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

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

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


######################################################################
# Once we have the threshold value we can assign the classes?;
######################################################################

test.df$ModelScores <- model.1$fitted.values;
test.df$classes <- ifelse(test.df$ModelScores>roc.specs$threshold,1,0);
```



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

test.df$ModelScores <- model.1$fitted.values;
test.df$classes <- ifelse(test.df$ModelScores>roc.specs$threshold,1,0);

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


# Let's create a proper confusion matrix
t <- table(test.df$PersonalLoan, test.df$classes);
# Compute row totals;
r <- apply(t,MARGIN=1,FUN=sum);
# Normalize confusion matrix to rates;
t/r

# Look at your confusion matrix and compare it to roc.specs.
# Do we see anything interesting?
# What values are on the diagonal?
# What values are on the off-diagonal?

#MODEL  2 CONFUSION MATRIX
test.df$ModelScores <- model.2$fitted.values;
test.df$classes <- ifelse(test.df$ModelScores>roc.specs$threshold,1,0);

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


# Let's create a proper confusion matrix
t <- table(test.df$PersonalLoan, test.df$classes);
# Compute row totals;
r <- apply(t,MARGIN=1,FUN=sum);
# Normalize confusion matrix to rates;
t/r



#MODEL  3 CONFUSION MATRIX
test.df$ModelScores <- model.3$fitted.values;
test.df$classes <- ifelse(test.df$ModelScores>roc.specs$threshold,1,0);

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


# Let's create a proper confusion matrix
t <- table(test.df$PersonalLoan, test.df$classes);
# Compute row totals;
r <- apply(t,MARGIN=1,FUN=sum);
# Normalize confusion matrix to rates;
t/r

```