Assignment #8

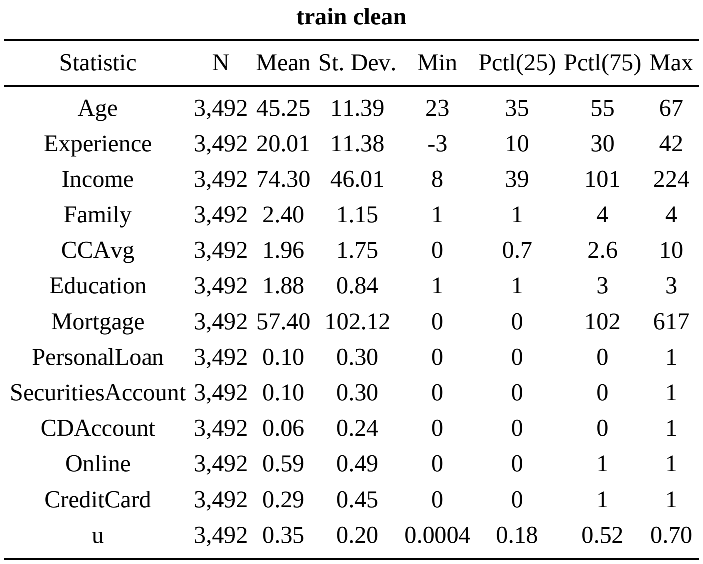
Alex Dantinne, MDSD410_SEC55

Introduction:

In this report, I will be using the UniversalBank.csv data set to build regression models for the response variable PersonalLoan.

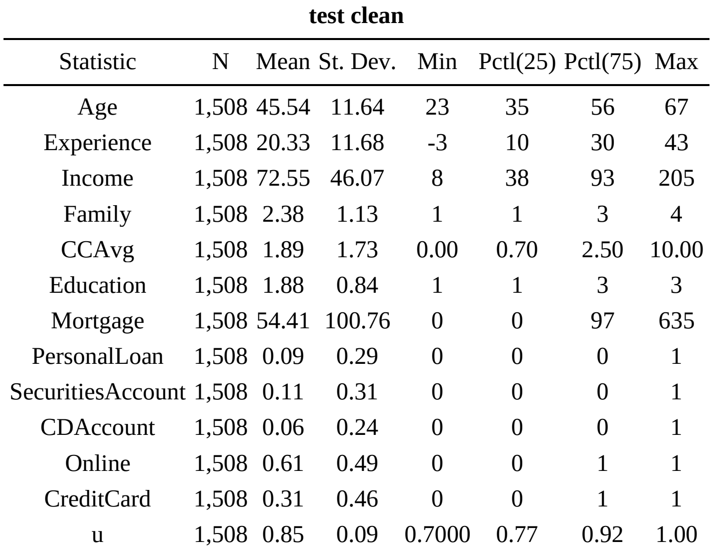
Section 1: Splitting, Cleaning, and EDA

For this report, I will be splitting the UniversalBank.csv data set into 70/30 training/test sets with a set seed of 12345. After the split, I dropped two predictor variables, 'ID' and 'ZIP.Code', from the original set for both the training set and test set. Below are summary tables for both the training set (train.clean) and the testing set (test.clean). There will be 3492 observations in the training set and 1508 observations in the testing set.



train clean

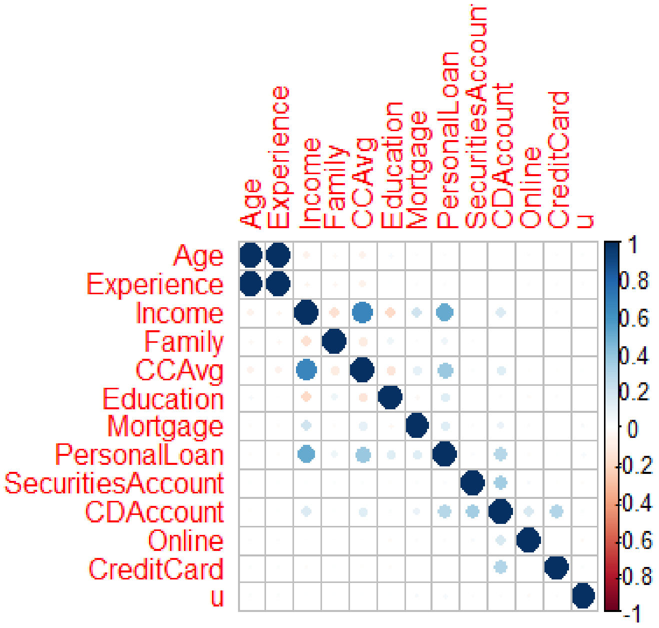
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Age	3,492	45.25	11.39	23	35	55	67
Experience	3,492	20.01	11.38	-3	10	30	42
Income	3,492	74.30	46.01	8	39	101	224
Family	3,492	2.40	1.15	1	1	4	4
CCAvg	3,492	1.96	1.75	0	0.7	2.6	10
Education	3,492	1.88	0.84	1	1	3	3
Mortgage	3,492	57.40	102.12	0	0	102	617
PersonalLoan	3,492	0.10	0.30	0	0	0	1
SecuritiesAccount	3,492	0.10	0.30	0	0	0	1
CDAccount	3,492	0.06	0.24	0	0	0	1
Online	3,492	0.59	0.49	0	0	1	1
CreditCard	3,492	0.29	0.45	0	0	1	1
u	3,492	0.35	0.20	0.0004	0.18	0.52	0.70



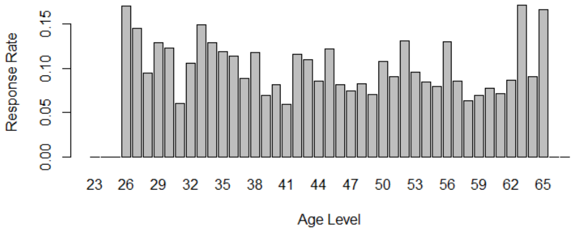
test clean

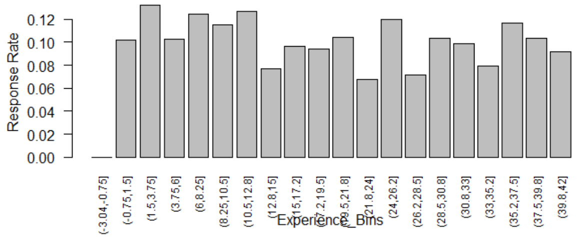
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Age	1,508	45.54	11.64	23	35	56	67
Experience	1,508	20.33	11.68	-3	10	30	43
Income	1,508	72.55	46.07	8	38	93	205
Family	1,508	2.38	1.13	1	1	3	4
CCAvg	1,508	1.89	1.73	0.00	0.70	2.50	10.00
Education	1,508	1.88	0.84	1	1	3	3
Mortgage	1,508	54.41	100.76	0	0	97	635
PersonalLoan	1,508	0.09	0.29	0	0	0	1
SecuritiesAccount	1,508	0.11	0.31	0	0	0	1
CDAccount	1,508	0.06	0.24	0	0	0	1
Online	1,508	0.61	0.49	0	0	1	1
CreditCard	1,508	0.31	0.46	0	0	1	1
u	1,508	0.85	0.09	0.7000	0.77	0.92	1.00

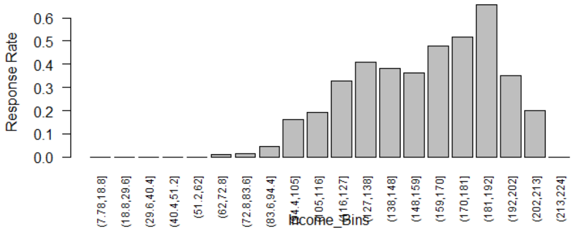
To further understand the training set, below is a corrplot. From the corrplot, there are three predictor variables(Income, CCAvg, and CDAccount) that have considerable correlation to the response variable PersonalLoan.

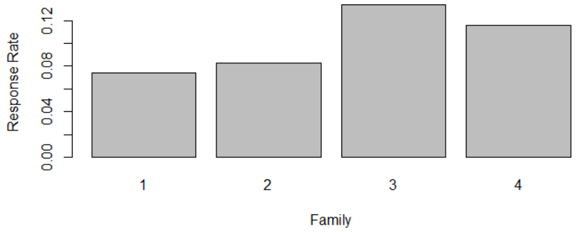


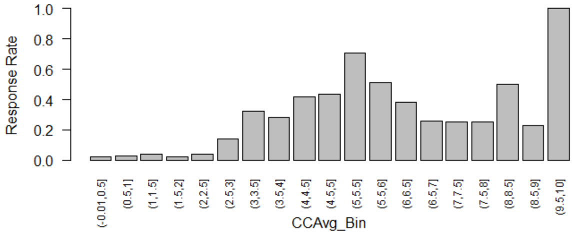
To further evaluate the training set, bar plots will be produced for each of the predictor variables with the same response variable PersonalLoan. Continuous variables will be discretized using bins.

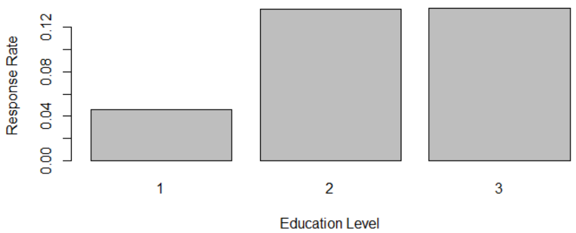


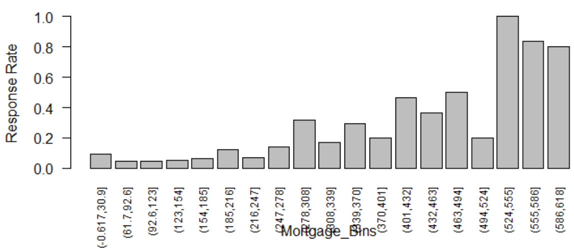


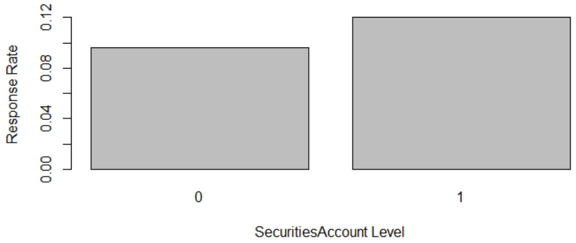


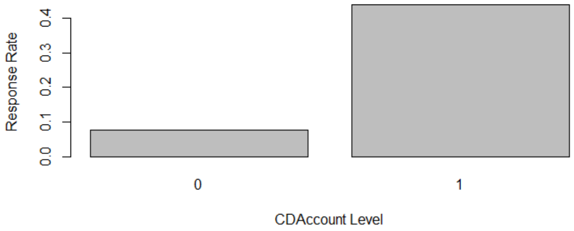


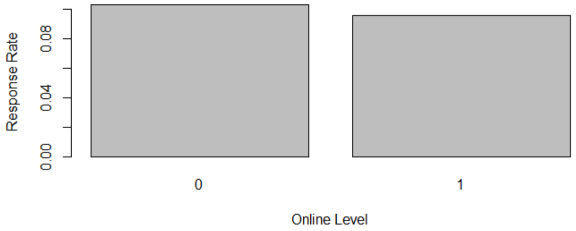

Response Rate
0.0
0.2
0.4
0.6
0.8
1.0
(-0.01,0.5]
(0.5,1]
(1,1.5]
(1.5,2]
(2,2.5]
(2.5,3]
(3,3.5]
(3.5,4]
(4,4.5]
(4.5,5]
(5,5.5]
(5.5,6]
(6,6.5]
(6.5,7]
(7,7.5]
(7.5,8]
(8,8.5]
(8.5,9]
(9.5,10]
CCAvg_Bin

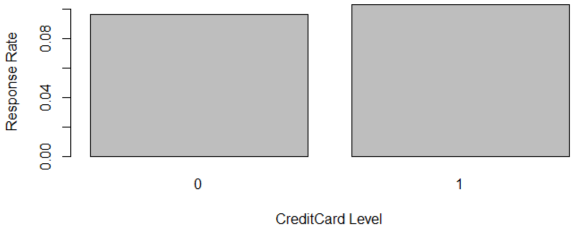

Response Rate
0.00
0.04
0.08
0.12
1
2
3
Education Level


Response Rate
0.0
0.2
0.4
0.6
0.8
1.0
(-0.617,30.9]
(61.7,92.6]
(92.6,123]
(123,154]
(154,185]
(185,216]
(216,247]
(247,278]
(278,308]
(308,339]
(339,370]
(370,401]
(401,432]
(432,463]
(463,494]
(494,524]
(524,555]
(555,586]
(586,618]
Mortgage_Bins


Response Rate
0.00
0.04
0.08
0.12
0
1
SecuritiesAccount Level


Response Rate
0.0
0.1
0.2
0.3
0.4
0
1
CDAccount Level


Response Rate
0.00
0.04
0.08
0
1
Online Level

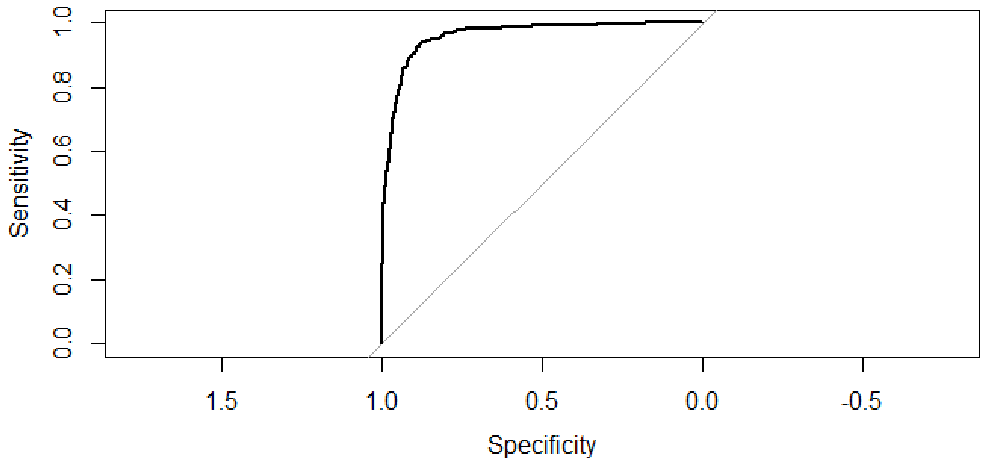

Response Rate
0.00
0.04
0.08
0
1
CreditCard Level

Section 2: Model Identification and Comparison

Next I will create Naïve model, model 1, as my base line model and create an additional two model for comparison. For each, I will produce a report, ROC curve, AUC metric, and confusion matrix.

model1.lm

	Dependent variable:
	PersonalLoan
Income	0.003***
	(0.0001)
CCAvg	0.01***
	(0.003)
CDAccount	0.27***
	(0.02)
factor(Education)2	0.16***
	(0.01)
factor(Education)3	0.16***
	(0.01)
Family	0.03***
	(0.004)
SecuritiesAccount	-0.05***
	(0.01)
Constant	-0.34***
	(0.01)
Observations	3,492
R^{2}	0.38
Adjusted R^{2}	0.38
Residual Std. Error	0.23 (df = 3484)
F Statistic	310.22*** (df = 7; 3484)
Note:	*p**p***p<0.01



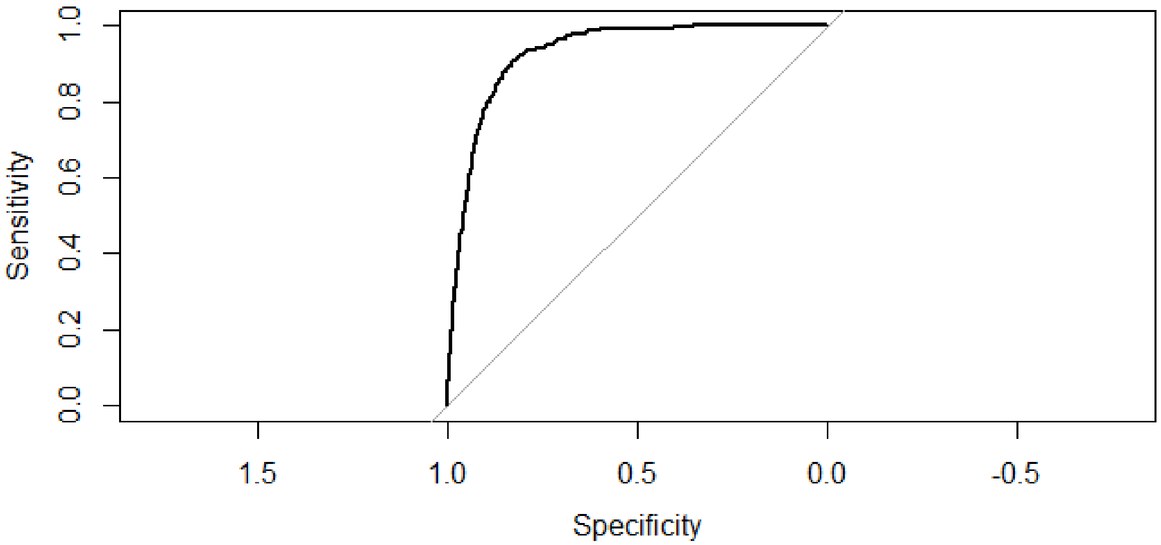
```
> auc.1
Area under the curve: 0.9605
```

Confusion Matrix using pROC

```
           0          1
  0 0.87801779 0.12198221
  1 0.06104651 0.93895349
```

stepwise.lm

	Dependent variable:
	PersonalLoan
Income	0.003***
	(0.0001)
Experience	0.001
	(0.0004)
Family	0.04***
	(0.004)
Constant	-0.26***
	(0.02)
Observations	3,492
R^2	0.27
Adjusted R^2	0.27
Residual Std. Error	0.25 (df = 3488)
F Statistic	433.69*** (df = 3; 3488)
Note:	*p**p***p<0.01



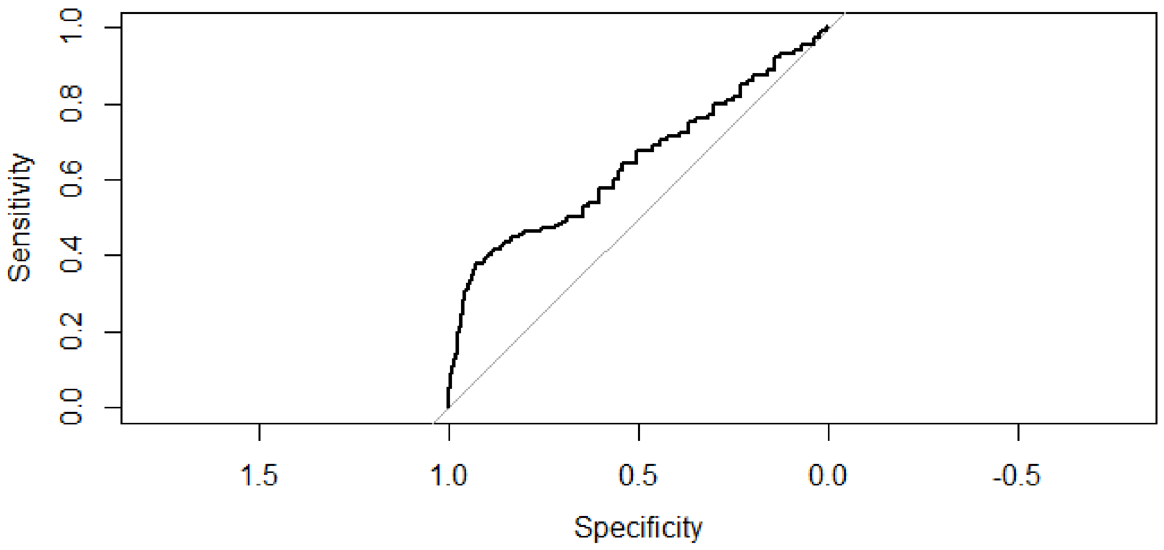
```
> auc.2
Area under the curve: 0.9289
```

Confusion Matrix using pROC

```
          0          1
  0 0.82941550 0.17058450
  1 0.09302326 0.90697674
```

junk.lm

	Dependent variable:
	PersonalLoan
Age	-0.0004
	(0.0004)
CDAccount	0.35^{***}
	(0.02)
Mortgage	0.0003^{***}
	(0.0000)
Constant	0.08^{***}
	(0.02)
Observations	3,492
R^2	0.09
Adjusted R^2	0.09
Residual Std. Error	0.28 (df = 3488)
F Statistic	121.47^{***} (df = 3; 3488)
Note:	*p**p***p<0.01



```
> auc.3
Area under the curve: 0.6443
```

Confusion Matric using pROC

```
           0          1
  0 0.92820839 0.07179161
  1 0.62209302 0.37790698
```

Of the three models above, model.1 preformed the best while junk model preformed the worst. The stepwise model preformed similar to model.1, but at a lesser rate. Adding additional predictor variables like CCAvg to the stepwise model may increase its performance. Junk model was created to explore age, CDs, and an existing mortgage as predictor variables. As the name reflects, it was the poorest preforming model. Comparing the AUC of each model, model.1 and stepwise had similar scores while junk had the lowest.

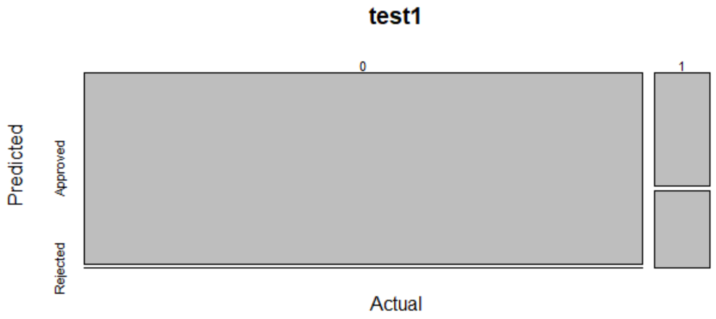
		AUC
1	AUC	0.96
2	AUC	0.93
3	AUC	0.64

Section 3: Predictive Accuracy

Finally, I will examine each of the created models for true positive rates and accuracy. Test 1 is for model1, test 2 is the stepwise model, and test 3 is the junk model. I will create a table showing confusion matrix for each as well as visual plot.

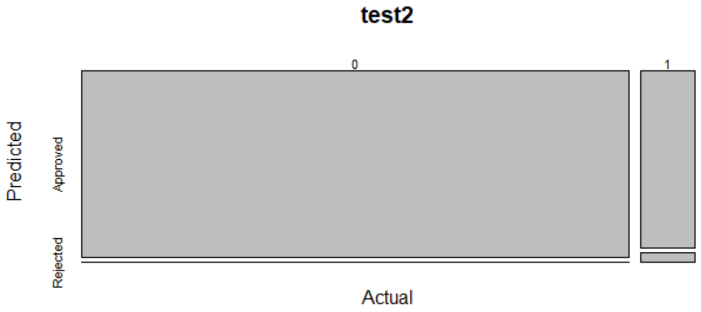
```
> test1
     Predicted
Actual Approved Rejected
    0     1371        1
    1       81       55
```

From the table above, the true positive rate is 94.4% and the accuracy is 94.8%.



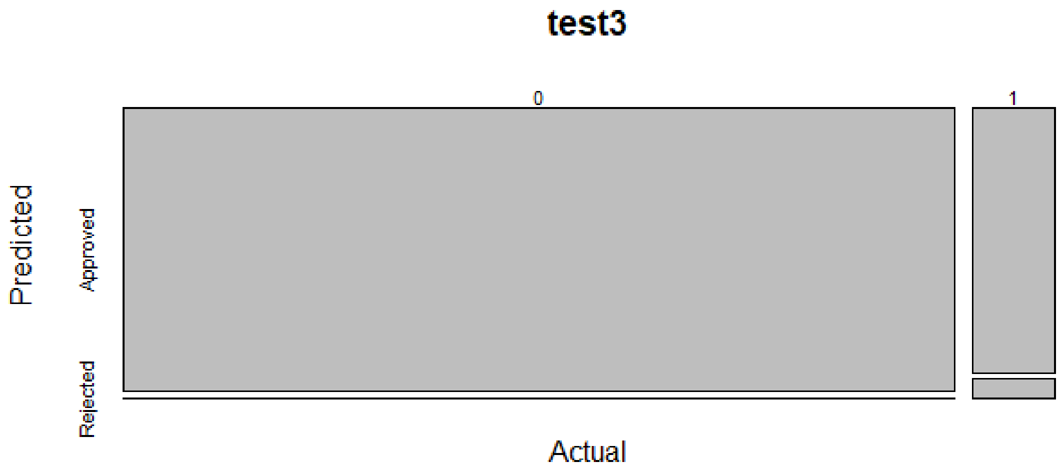
```
> test2
     Predicted
Actual Approved Rejected
    0     1369        3
    1      129        7
```

From the table above, the true positive rate is 91.4% and the accuracy is 91.2%.



```
> test3
      Predicted
Actual Approved Rejected
     0     1369        3
     1      127        9
```

From the table above, the true positive rate is 91.5% and the accuracy is 91.4%.



Again, model 1 preformed the best model out-of-sample.

Conclusion:

To sum, this report used the UniversalBank.csv data set to build logistic regression models for binary classification and explore the response variable PersonLoan. Of the built models above, model1 perform the best in and out of sample. Going forward, I would choose model1 and its predictor variables over the other models built for this report.

Code:

```
# Assignment_8_DantinneAlex.R
# Alex Dantinne
# 11.14.21


my.path <- 'C:\\Users\\alex.dantinne\\Documents\\Northwestern\\6_Fall 2021\\MSDS 410\\Week 8\\'
my.file <- paste(my.path,'UniversalBank.csv',sep='');
my.data <- read.csv(my.file,header=TRUE);

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

drop.list <-c('ID', 'ZIP.Code')
my.data <-my.data[,!(names(my.data)%in%drop.list)];
my.data <-my.data[complete.cases(my.data), ]
str(my.data)

library(MASS)
library(stargazer)
out.path <- 'C:\\Users\\alex.dantinne\\Documents\\Northwestern\\6_Fall 2021\\MSDS 410\\Week 7';
file.name <- 'mydata.html';
stargazer(my.data, type=c('html'),out=paste(out.path,file.name,sep=''),
        title=c('my.data'),
        align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

corrplot(cor(my.data))
plot(my.data)

##### split
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);

drop.list <-c('ID', 'ZIP.Code')
train.clean <-train.df[,!(names(my.data)%in%drop.list)];
train.clean <-train.clean[complete.cases(train.clean), ]
str(train.clean)

drop.list <-c('ID', 'ZIP.Code')
test.clean <-test.df[,!(names(my.data)%in%drop.list)];
test.clean <-test.clean[complete.cases(test.clean), ]
str(test.clean)

library(MASS)
library(stargazer)
out.path <- 'C:\\Users\\alex.dantinne\\Documents\\Northwestern\\6_Fall 2021\\MSDS 410\\Week 7';
file.name <- 'train clean.html';
stargazer(train.clean, type=c('html'),out=paste(out.path,file.name,sep=''),
        title=c('train clean'),
        align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

out.path <- 'C:\\Users\\alex.dantinne\\Documents\\Northwestern\\6_Fall 2021\\MSDS 410\\Week 7';
file.name <- 'test clean.html';
stargazer(test.clean, type=c('html'),out=paste(out.path,file.name,sep=''),
        title=c('test clean.df'),
        align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

library(corrplot)
corrplot(cor(train.clean))
boxplot(train.clean)


######################################################################
# Response rates for discrete variables
######################################################################

response.Age <- aggregate(train.clean$PersonalLoan,
                          by=list(Age=train.clean$Age),
                          FUN=mean);

barplot(height=response.Age$x,names.arg=response.Age$Age,
    xlab='Age Level',ylab='Response Rate')


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

barplot(height=response.Family$x,names.arg=response.Family$Family,
    xlab='Family',ylab='Response Rate')


response.Education <- aggregate(train.clean$PersonalLoan,
                      by=list(Education=train.clean$Education),
                      FUN=mean);

barplot(height=response.Education$x,names.arg=response.Education$Education,
           xlab='Education Level',ylab='Response Rate')


response.SecuritiesAccount <- aggregate(train.clean$PersonalLoan,
               by=list(SecuritiesAccount=train.clean$SecuritiesAccount),
               FUN=mean);

barplot(height=response.SecuritiesAccount$x,names.arg=response.SecuritiesAccount$SecuritiesAccount,
    xlab='SecuritiesAccount Level',ylab='Response Rate')

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

barplot(height=response.CDAccount$x,names.arg=response.CDAccount$CDAccount,
    xlab='CDAccount Level',ylab='Response Rate')

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

barplot(height=response.Online$x,names.arg=response.Online$Online,
    xlab='Online Level',ylab='Response Rate')

response.CreditCard <- aggregate(train.clean$PersonalLoan,
              by=list(CreditCard=train.clean$CreditCard),
              FUN=mean);

barplot(height=response.CreditCard$x,names.arg=response.CreditCard$CreditCard,
    xlab='CreditCard Level',ylab='Response Rate')

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

train.clean$Experience_Bins <- cut(train.clean$Experience,breaks=20)
table(train.clean$Experience_Bins)

response.Experience_Bins <- aggregate(train.clean$PersonalLoan,
                by=list(Experience_Bins=train.clean$Experience_Bins),
                FUN=mean);

barplot(height=response.Experience_Bins$x,names.arg=response.Experience_Bins$Experience_Bins,
    xlab='Experience_Bins',ylab='Response Rate',las=2,cex.names=0.75)


train.clean$Income_Bins <- cut(train.clean$Income,breaks=20)
table(train.clean$CCAvg_Bins)

response.Income_Bins <- aggregate(train.clean$PersonalLoan,
                by=list(Income_Bins=train.clean$Income_Bins),
                FUN=mean);

barplot(height=response.Income_Bins$x,names.arg=response.Income_Bins$Income_Bins,
    xlab='Income_Bins',ylab='Response Rate',las=2,cex.names=0.75)


train.clean$CCAvg_Bins <- cut(train.clean$CCAvg,breaks=20)
table(train.clean$CCAvg_Bins)
```

```
response.CCAvg_Bins <- aggregate(train.clean$PersonalLoan,
                                  by=list(CCAvg_Bins=train.clean$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)


train.clean$Mortgage_Bins <- cut(train.clean$Mortgage,breaks=20)
table(train.clean$Mortgage_Bins)

response.Mortgage_Bins <- aggregate(train.clean$PersonalLoan,
                            by=list(Mortgage_Bins=train.clean$Mortgage_Bins),
                            FUN=mean);

barplot(height=response.Mortgage_Bins$x,names.arg=response.Mortgage_Bins$Mortgage_Bins,
   xlab='Mortgage_Bins',ylab='Response Rate',las=2,cex.names=0.75)


######################################################################
# Fit a Naive Model
######################################################################

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

file.name <- 'model1.html';
stargazer(model.1, type=c('html'),out=paste(out.path,file.name,sep=''),
    title=c('model1.lm'),
    align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

Income.lm <- lm(PersonalLoan ~ Income+Experience+Family,data=train.clean);
summary(Income.lm)
stepwise.lm <- stepAIC(object=Income.lm, direction=c('both'));
summary(stepwise.lm)
file.name <- 'stepwise.html';
stargazer(stepwise.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
    title=c('stepwise.lm'),
    align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

junk.lm <- lm(PersonalLoan ~ Age + CDAccount+Mortgage, data=train.clean)
summary(junk.lm)
file.name <- 'junk.html';
stargazer(junk.lm, type=c('html'),out=paste(out.path,file.name,sep=''),
    title=c('junk.lm'),
    align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

######################################################################


library(pROC)


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

auc.1 <- auc(roc.1);
auc.1

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

auc.2 <- auc(roc.2);
auc.2

roc.3 <- roc(response=train.clean$PersonalLoan, predictor=junk.lm$fitted.values)
print(roc.3)
plot(roc.3)

auc.3 <- auc(roc.3);
auc.3


########################################################################
# 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
)

roc2.specs <- coords(roc=roc.2,x=c('best'),
input=c('threshold'),
ret=c('threshold','specificity','sensitivity'),
as.list=TRUE
)

roc3.specs <- coords(roc=roc.3,x=c('best'),
input=c('threshold'),
ret=c('threshold','specificity','sensitivity'),
as.list=TRUE
)

auc <- matrix(c(auc.1, auc.2, auc.3),ncol=1,byrow=TRUE)
colnames(auc) <- c("Auc")
rownames(auc) <- c("1","2","3")
auc <- as.table(auc)
auc
file.name <- 'Auc.html';
stargazer(auc, type=c('html'),out=paste(out.path,file.name,sep=''),
        title=c('Auc.lm'),
        align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE)

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

table(train.clean$PersonalLoan, train.clean$classes)

t <- table(train.clean$PersonalLoan, train.clean$classes);
r <- apply(t,MARGIN=1,FUN=sum);
t/r

train.clean$ModelScores <- stepwise.lm$fitted.values;
train.clean$classes <- ifelse(train.clean$ModelScores>roc2.specs$threshold,1,0);

table(train.clean$PersonalLoan, train.clean$classes)

t2 <- table(train.clean$PersonalLoan, train.clean$classes);
r2 <- apply(t,MARGIN=1,FUN=sum);
t2/r2

train.clean$ModelScores <- junk.lm$fitted.values;
train.clean$classes <- ifelse(train.clean$ModelScores>roc3.specs$threshold,1,0);

table(train.clean$PersonalLoan, train.clean$classes)

t3 <- table(train.clean$PersonalLoan, train.clean$classes);
r3 <- apply(t,MARGIN=1,FUN=sum);
t3/r3

#################################
library(caret)

model1.test <- predict.glm(model.1,newdata=test.clean);
act1<- test.clean$PersonalLoan
pred.test1<-factor(model1.test >0.5, levels = c(FALSE, TRUE),labels = c("Approved", "Rejected"))
test1 <- table(act1, pred.test1,dnn = c("Actual", "Predicted"))
test1
plot(test1)

stepwise.test <- predict.glm(stepwise.lm,newdata=test.clean);
act2<- test.clean$PersonalLoan
```

```
pred.test2<-factor(stepwise.test >0.5, levels = c(FALSE, TRUE),labels = c("Approved", "Rejected"))
test2 <- table(act2, pred.test2,dnn = c("Actual", "Predicted"))
test2
plot(test2)

junk.test <- predict.glm(junk.lm,newdata=test.clean);
act3<- test.clean$PersonalLoan
pred.test3<-factor(junk.test >0.5, levels = c(FALSE, TRUE),labels = c("Approved", "Rejected"))
test3 <- table(act3, pred.test3,dnn = c("Actual", "Predicted"))
test3
plot(test3)
```