Ethan Poris - 250998471
Feb 5, 2021

Having evaluated the model commissioned to use only information on whether a customer is an active user of online banking serves and whether the customer has a credit card issued by the bank, I recommend against implementing it to target consumers as it is unable to predict who will accept offers for personal loans better than chance. Instead, I recommend using a model that includes either more predictors, or at least more relevant predictors.

The model was constructed by training it on 2/3 the data we have and testing the resultant model on the remaining third. One of the major issues with the model is that it generates extremely similar probabilities of accepting the personal loan for all individuals, with a minimum probability of 0.088 and a maximum of 0.105. This indicates that the model essentially sees no difference between people who will and will not accept the loan based on the included predictors.

Further illustrating this point is the model's confusion matrix that uses the mean probability of acceptance as a criterion for classifying who will accept the loan. As seen in Exhibit 1, the test set contained 2000 individuals, 199 of which actually accepted personal loans. Of those 199, the most the model was able to identify was 51, missing 148 and incorrectly predicting that 514 who did not accept the personal loan would. This means that should the model have been used to target people within this testing group, it would have successfully converted 9.0% (51/565) of the people the promotion was sent to, which is essentially the same as the 10% we would convert if we simply sent the promotion to all individuals in the data set (199/2000).

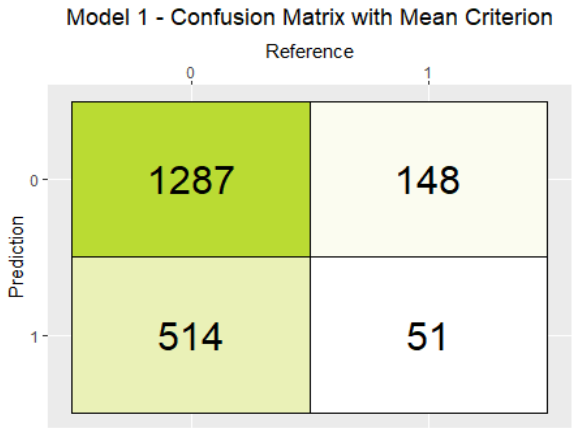
There are two major factors I believe could be are responsible for the poor performance of the model. The first is that the predictors used to create it not strongly enough related to whether someone will accept a personal loan to construct a useful model. This could be solved by simply including more and/or a different set of predictor variables. The second is that there are far more people who did not accept the personal loan than did, resulting in a scarcity of the information required to construct an accurate model. This will likely be solved as the promotion is sent to more people and more data collected.

To answer which of these factors is most likely, I made a second model that included more of the variables in the dataset to predict who will accept the personal loan offer. This model included the predictors of, "Age," "Experience," "Income," "Family," and "CCAvg," in addition to those already in the model. This second model is far more accurate at predicting who will accept personal loan offer (See Exhibit 2), which suggests that while information from more people would likely make the model better, the biggest issue with the original model is that it did not include the most *relevant* information available for accomplishing our goal.

To add further evidence that it is the relevance rather than the quantity of the information that caused the poor performance of the proposed model, I created a third model that still only uses two variables to predict who will accept the personal loan: age and income. Again, this model vastly outperforms the proposed one, nearly performing as well as the second model which included age, experience, income, family, ccavg, online, and credit card (See Exhibit 3).

In conclusion, I once again must recommend not to use the proposed model for this initiative as it performs at chance levels, most likely due to the irrelevance of the predictors it uses. Instead, I would recommend using a model with either more predictors and/or, more critically, variables that are more relevant to whether someone will accept our personal loan prediction.

Exhibit 1 – Confusion Matrix and ROC Curve for Commissioned Model that uses Online and CreditCard



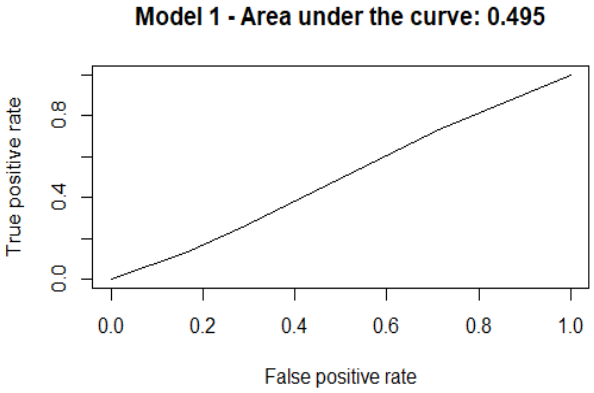
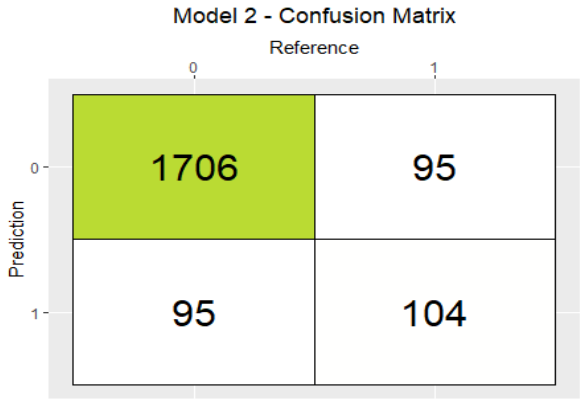


Exhibit 2 – Confusion Matrix and ROC Curve for Model 2 which uses predictors Age, Experience, Income, Family, CCAvg, Online and CreditCard



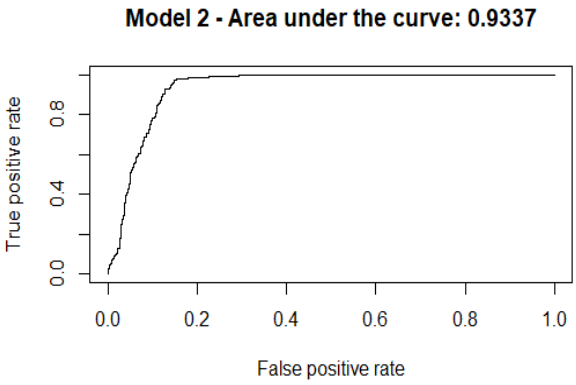
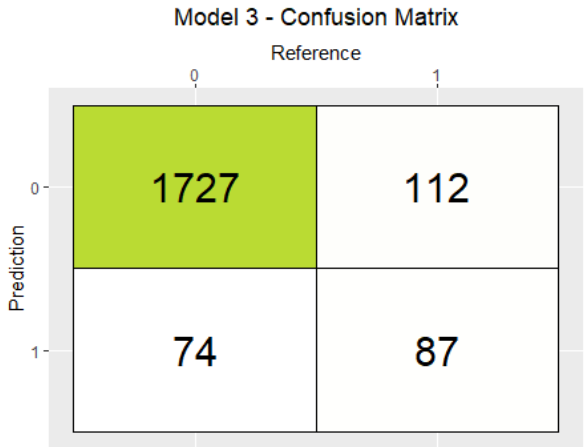


Exhibit 3 – Confusion Matrix and ROC Curve for Model 3 which uses predictors Age and Income only



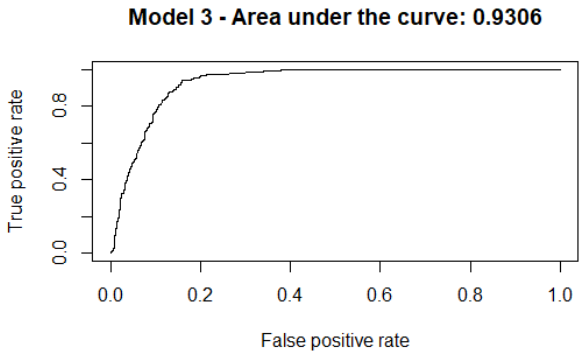


Exhibit 4 – Code

```
# The file Bank.csv contains data on 5000 customers. For this exercise, focus only
# on two two predcitors: Online (whetehr or not a customer is an active user of
# online banking services) and Credit Card (does the customer have a credit card
# issued by the bank). Partition the data into training (60%) and validation (40%).
# The goal is to predict customer response to a the personal loan campaign.
#
# Use Naïve Bayes and compute the confusion matrix and ROC curve and comment.
# The beginning of your report should contain a managerial conclusion.

### Model 1 – Commissioned model that uses Online and CreditCard to predict Personal.Loan ###
# Initialize relevant packages and load data
install.packages("Rtools")
install.packages("yardstick")
library(yardstick)
library(e1071)
library(caret)
library(ROCR)
library(ggplot2)
library(dplyr)
library(tidyverse)
setwd('C:/Users/Ethan/Documents/Ivey/T2/Big Data Analytics')
bank <- read.csv('./Assignment 2/Bank-1.csv', header=TRUE)
head(bank)

# Select Personal Loans, Online and Credit card as variables and split into training and test sets
selected.var <- c(10, 13, 14)
train.index <- sample(c(1:dim(bank)[1]), dim(bank)[1]*0.6)
train.df <- bank[train.index, selected.var]
valid.df <- bank[-train.index, selected.var]

# Train NB model to predict Personal Loans using Online and Credit Cards
bank.nb <- naiveBayes(as.factor(Personal.Loan) ~ ., data = train.df)
bank.nb

# Use NB to predict P(accept loan) for people in test set; assign to 'pred.prob'
pred.prob <- predict(bank.nb, newdata = valid.df, type = "raw")
pred.prob

# Use NB to predict class of people in test set (1 = accept loan); assign to 'pred.class'
pred.class <- predict(bank.nb, newdata = valid.df)
pred.class

# assign 'df' four columns: actual class, pred class, prob(decline), prob(accept)
df <- data.frame(actual = valid.df$Personal.Loan, predicted = pred.class, pred.prob)
```

```
head(df)

#Classification uses .5 criterion, classification 2 is for playing with criterion
# Classification vector of 1s if pred.prob's P(accept) > criterion, 0 if < criterion
actual <- valid.df$Personal.Loan
classification <- ifelse(pred.prob[,2]>.5,1,0)
classification2 <- ifelse(pred.prob[,2]>mean(pred.prob[,2]),1,0)

summary(pred.prob[,2])

# Make confusion matrix
cm1 <- confusionMatrix(as.factor(classification), as.factor(actual))
cm2 <- confusionMatrix(as.factor(classification2), as.factor(actual))

# Draw confusion matrices
cm1$table %>%
  data.frame() %>%
  group_by(Reference) %>%
  mutate(total = sum(Freq)) %>%
  ungroup() %>%
  ggplot(aes(Reference, reorder(Prediction, desc(Prediction)), fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), size = 8) +
  scale_fill_gradient(low = "white", high = "#badb33") +
  scale_x_discrete(position = "top") +
  ggtitle("Confusion Matrix with Base Criterion") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("Prediction") +
  geom_tile(color = "black", fill = "black", alpha = 0)

cm2$table %>%
  data.frame() %>%
  group_by(Reference) %>%
  mutate(total = sum(Freq)) %>%
  ungroup() %>%
  ggplot(aes(Reference, reorder(Prediction, desc(Prediction)), fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), size = 8) +
  scale_fill_gradient(low = "white", high = "#badb33") +
  scale_x_discrete(position = "top") +
  ggtitle("Model 1 - Confusion Matrix with Mean Criterion") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("Prediction") +
  geom_tile(color = "black", fill = "black", alpha = 0)

# make predObj, rocObj, and aucObj to make ROC curve
predObj <- prediction(pred.prob[,2], valid.df$Personal.Loan)
rocObj = performance(predObj, measure="tpr", x.measure="fpr")
```

```
aucObj = performance(predObj, measure="auc")

#plot ROC curve
plot(rocObj, main = paste("Model 1 - Area under the curve:", round(aucObj@y.values[[1]], 4)))

######
## NB model 2 - include more variables to see if can make better model for predicting Personal.Loan ##
# Includes Age, Experience, Income, Family, CCAvg, Online and CreditCard

# Select Personal Loans, Online and Credit card as variables and split into training and test sets
selected.var <- c(10, 2, 3, 4, 6, 7, 13, 14)
train.df <- bank[train.index, selected.var]
valid.df <- bank[-train.index, selected.var]

# Train NB model to predict Personal Loans using Online and Credit Cards
bank.nb <- naiveBayes(as.factor(Personal.Loan) ~ ., data = train.df)

# Use NB to predict P(accept loan) for people in test set; assign to 'pred.prob'
pred.prob <- predict(bank.nb, newdata = valid.df, type = "raw")
pred.prob

# Use NB to predict class of people in test set (1 = accept loan); assign to 'pred.class'
pred.class <- predict(bank.nb, newdata = valid.df)
pred.class

# assign 'df' four columns: actual class, pred class, prob(decline), prob(accept)
df <- data.frame(actual = valid.df$Personal.Loan, predicted = pred.class, pred.prob)
head(df)

summary()
#Classification uses .5 criterion, classification 2 is for playing with criterion
# Classification vector of 1s if pred.prob's P(accept) > criterion, 0 if < criterion
actual <- valid.df$Personal.Loan
classification <- ifelse(pred.prob[,2]>.5,1,0)

# Make confusion matrix
cm1 <- confusionMatrix(as.factor(classification), as.factor(actual))

# Draw confusion matrices
cm1$table %>%
  data.frame() %>%
  group_by(Reference) %>%
  mutate(total = sum(Freq)) %>%
  ungroup() %>%
  ggplot(aes(Reference, reorder(Prediction, desc(Prediction)), fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), size = 8) +
  scale_fill_gradient(low = "white", high = "#badb33") +
```

```
  scale_x_discrete(position = "top") +
  ggtitle("Model 2 - Confusion Matrix") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("Prediction") +
  geom_tile(color = "black", fill = "black", alpha = 0)

# make predObj, rocObj, and aucObj to make ROC curve
predObj <- prediction(pred.prob[,2], valid.df$Personal.Loan)
rocObj = performance(predObj, measure="tpr", x.measure="fpr")
aucObj = performance(predObj, measure="auc")

#plot ROC curve
plot(rocObj, main = paste("Model 2 - Area under the curve:", round(aucObj@y.values[[1]], 4)))

#######
### NB model 3 - Use Age and Income to predict Personal.Loan ###

# Select Personal Loans, Online and Credit card as variables and split into training and test sets
selected.var <- c(10, 2, 4)
train.df <- bank[train.index, selected.var]
valid.df <- bank[-train.index, selected.var]

# Train NB model to predict Personal Loans using Online and Credit Cards
bank.nb <- naiveBayes(as.factor(Personal.Loan) ~ ., data = train.df)

# Use NB to predict P(accept loan) for people in test set; assign to 'pred.prob'
pred.prob <- predict(bank.nb, newdata = valid.df, type = "raw")
pred.prob

# Use NB to predict class of people in test set (1 = accept loan); assign to 'pred.class'
pred.class <- predict(bank.nb, newdata = valid.df)
pred.class

# assign 'df' four columns: actual class, pred class, prob(decline), prob(accept)
df <- data.frame(actual = valid.df$Personal.Loan, predicted = pred.class, pred.prob)
head(df)

summary()
#Classification uses .5 criterion, classification 2 is for playing with criterion
# Classification vector of 1s if pred.prob's P(accept) > criterion, 0 if < criterion
actual <- valid.df$Personal.Loan
#classification <- ifelse(pred.prob[,2]>.3,1,0)
classification <- ifelse(pred.prob[,2]>.5,1,0)

# Make confusion matrix
cm1 <- confusionMatrix(as.factor(classification), as.factor(actual))

# Draw confusion matrices
```

```
cm1$table %>%
  data.frame() %>%
  group_by(Reference) %>%
  mutate(total = sum(Freq)) %>%
  ungroup() %>%
  ggplot(aes(Reference, reorder(Prediction, desc(Prediction)), fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), size = 8) +
  scale_fill_gradient(low = "white", high = "#badb33") +
  scale_x_discrete(position = "top") +
  ggtitle("Model 3 - Confusion Matrix") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ylab("Prediction") +
  geom_tile(color = "black", fill = "black", alpha = 0)

# make predObj, rocObj, and aucObj to make ROC curve
predObj <- prediction(pred.prob[,2], valid.df$Personal.Loan)
rocObj = performance(predObj, measure="tpr", x.measure="fpr")
aucObj = performance(predObj, measure="auc")

#plot ROC curve
plot(rocObj, main = paste("Model 3 - Area under the curve:", round(aucObj@y.values[[1]], 4)))
```