LoanDefaultROC

Hannah Aguirre

2024-12-12

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(ROCR)
df <- read.csv("~/Downloads/Loan_default.csv", stringsAsFactors = T)
df <- df[,-1] #Drop ID as it is not necessary
df$Education <- factor(df$Education,levels = c("High School","Bachelor's","Master's","PhD")) #Reorder L
df$MaritalStatus <- factor(df$MaritalStatus,levels = c("Single","Married","Divorced"))
df$Default <- factor(df$Default)
```

```
set.seed(123)
idx <- sample(1:225694,replace = F, size = 29653)
balance <- df[df$Default == 0,]
df <- rbind(df[df$Default == 1,],balance[idx,]) #Balance dataset
```

```
set.seed(123)
index <- createDataPartition(df$Default, p=.8, list=FALSE, times=1)
train <- df[index,]
test <- df[-index,] #Create train and test split
```

```
logreg <- glm(Default~.,family = 'binomial',data = train)
probs1 <- predict(logreg, test, type = 'response')

library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
x <- model.matrix(Default~., data = train)[,-1]
glmmod <- glmnet(x, y=as.factor(train$Default), alpha=1, family="binomial", lambda = 0.03)
xtest <- model.matrix(Default~., data = test)[,-1]
probs2 <- predict(glmmod, newx = xtest, type = 'response')
pred_m2 <- ROCR::prediction(probs2, test$Default)
roc_curve2 <- ROCR::performance(pred_m2, "tpr","fpr")
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
rf_model = randomForest(Default~., data = train, ntree = 500, mtry = 2, importance = TRUE)
```

```r
probability <- predict(rf_model, test, "prob")
probability <- probability[,"1"]
library(ROCR)

pred_m3 <- ROCR::prediction(probability, test$Default)
roc_curve3 <- ROCR::performance(pred_m3, "tpr","fpr")
```

```r
library(neuralnet)
```

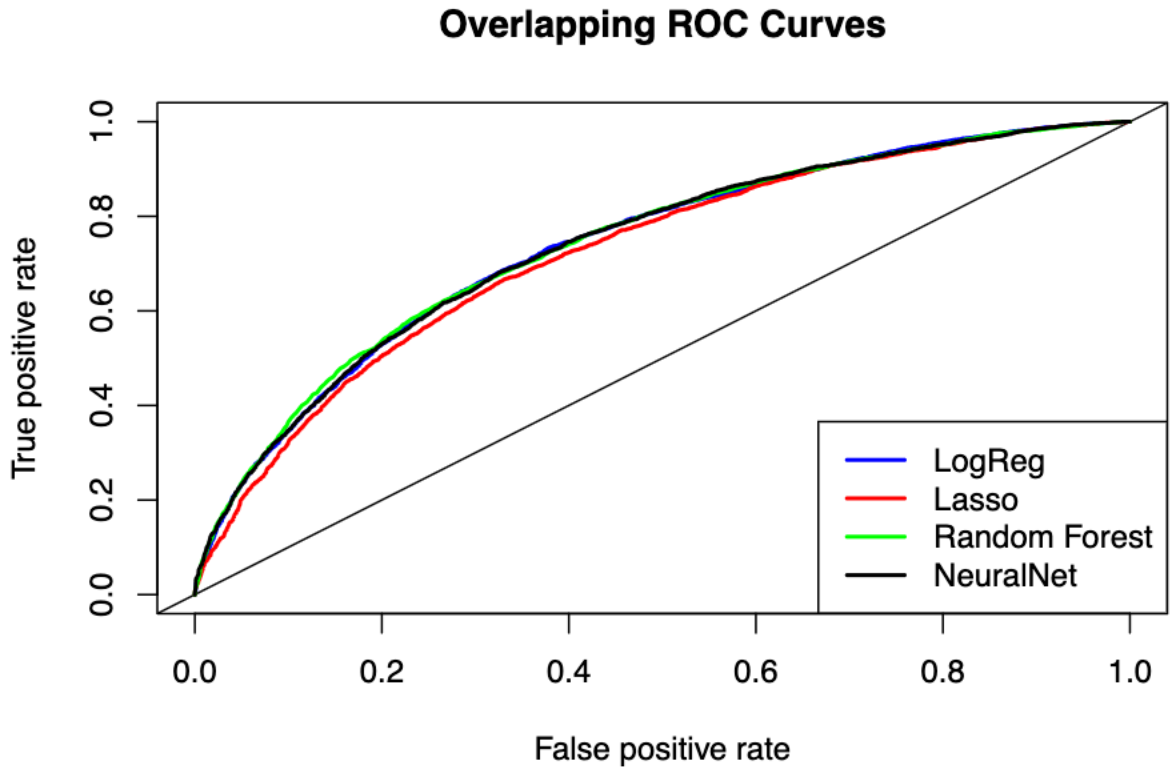
```
##
## Attaching package: 'neuralnet'

## The following object is masked from 'package:ROCR':
##
##     prediction
```

```r
selected_features = c("Age", "Income", "LoanAmount", "MonthsEmployed", "InterestRate")
train_nn = train[, selected_features]
train_nn$Default = as.numeric(train$Default)
maxs = apply(train_nn, 2, max)
mins = apply(train_nn, 2, min)
scaled_train = as.data.frame(base::scale(train_nn, center = mins, scale = maxs - mins))
formula = as.formula(paste("Default ~", paste(selected_features, collapse = "+")))
set.seed(123)
nn = neuralnet(
  formula,
  data = scaled_train,
  hidden = c(3),
  act.fct = "logistic",
  linear.output = FALSE,
  stepmax = 5e+05
)
```

```
scaled_test = as.data.frame(scale(test[, selected_features], center = mins[selected_features], scale = r
scaled_test$Default = test$Default
nn_predictions = compute(nn, scaled_test[, selected_features])
probs4 = nn_predictions$net.result
```

```
pred_m4 <-ROCR::prediction(probs4, test$Default)
roc_curve4 <- ROCR::performance(pred_m4, "tpr","fpr")
```

```
pred_m1 <- ROCR::prediction(probs1, test$Default)
roc_curve1 <- ROCR::performance(pred_m1, "tpr","fpr")
plot(roc_curve1, col = "blue", lwd = 2, main = "Overlapping ROC Curves")
abline(0, 1)
plot(roc_curve2, col="red", lwd = 2, add = T)
plot(roc_curve3, col="green", lwd = 2, add = T)
plot(roc_curve4, col="black", lwd = 2, add = T)
legend("bottomright",
       legend = c("LogReg", "Lasso", "Random Forest", "NeuralNet"),
       col = c("blue", "red", "green", "black"),
       lwd = 2)
```



Results and Conclusions: Comparing the ROC curves of each of the models, we can see that their curves almost completely overlap, and their area under the curve is almost equivalent, with Random Forest being just barely higher than the rest. The accuracy of each model is also almost equivalent for all four models, which allows us to choose the most parsimonious model. Since the Lasso and Neural Network each have five predictors, compared to the twenty four, we could choose either model for simplicity and accuracy, but will

choose the Neural Network for it's slight improvement in accuracy and area under the curve. Using these models we found that the most important predictors are Income, Interest Rate, Age, Loan Amount, and Months Employed. The neural network showed us that Income has the largest negative impact on defaulting, while Interest Rate has the largest positive impact. Some of our assumptions were that income, credit score, dependents, and education would have a larger impact on default on loans. However, we are able to see that, while income is an important predictor of loan default, credit score, dependents, and education are not.