Intro to Data Science HW 8

Copyright Jeffrey Stanton, Jeffrey Saltz, and Jasmina Tacheva

```
# Enter your name here: Benjamin Tisinger
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
# Help from Github here -> https://github.com/Enno-Victor/R-Machine-Learning-by-Example-by-Raghav-Bali/blob/master/Credit%20Risk%20Project-%20Classification%20Exp.R
# Tree Map Help Here -> https://www.statmethods.net/advstats/cart.html
# Tree Map Help Also Here -> https://www.datatechnotes.com/2018/04/classification-with-bagging-treebag.html
```

Supervised learning means that there is a **criterion one is trying to predict**. The typical strategy is to **divide data** into a **training set** and a **test set** (for example, **two-thirds training** and **one-third test**), train the model on the training set, and then see how well the model does on the test set.

Support vector machines (SVM) are a highly flexible and powerful method of doing **supervised machine learning**.

Another approach is to use **partition trees (rpart)**

In this homework, we will use another banking dataset to train an SVM model, as well as an rpart model, to **classify potential borrowers into 2 groups of credit risk** – **reliable borrowers** and **borrowers posing a risk**. You can learn more about the variables in the dataset here: https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29 (https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29)

This kind of classification algorithms is used in many aspects of our lives – from credit card approvals to stock market predictions, and even some medical diagnoses.

Part 1: Load and condition the data

A. Read the contents of the following .csv file into a dataframe called **credit**:

https://intro-datascience.s3.us-east-2.amazonaws.com/GermanCredit.csv (https://intro-datascience.s3.us-east-2.amazonaws.com/GermanCredit.csv)

You will also need to install( ) and library( ) several other libraries, such as **kernlab** and **caret**.

```r
library(kernlab)
library(caret)
```

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

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:kernlab':
##
##     alpha
```

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

```
library(tidyverse)
```

```
## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ✔ purrr   0.3.5
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ ggplot2::alpha() masks kernlab::alpha()
## ✖ purrr::cross()   masks kernlab::cross()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ✖ purrr::lift()    masks caret::lift()
```

```
library(e1071)
```

```
credit <- read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/GermanCredit.csv")
head(credit,3)
```

```
##                status duration                          credit_history
## 1        ... < 100 DM        6  critical account/other credits existing
## 2   0 <= ... < 200 DM       48 existing credits paid back duly till now
## 3 no checking account       12  critical account/other credits existing
##               purpose amount                    savings employment_duration
## 1 domestic appliances   1169 unknown/no savings account      ... >= 7 years
## 2 domestic appliances   5951               ... < 100 DM  1 <= ... < 4 years
## 3          retraining   2096               ... < 100 DM  4 <= ... < 7 years
##   installment_rate                personal_status_sex other_debtors
## 1                4                      male : single          none
## 2                2 female : divorced/separated/married          none
## 3                2                      male : single          none
##   present_residence    property age other_installment_plans housing
## 1                 4 real estate  67                    none     own
## 2                 2 real estate  22                    none     own
## 3                 3 real estate  49                    none     own
##   number_credits                       job people_liable telephone
## 1              2 skilled employee/official             1       yes
## 2              1 skilled employee/official             1        no
## 3              1      unskilled - resident             2        no
##   foreign_worker credit_risk
## 1            yes           1
## 2            yes           0
## 3            yes           1
```

B. Which variable contains the outcome we are trying to predict, **credit risk**? For the purposes of this analysis, we will focus only on the numeric variables and save them in a new dataframe called **cred**:

```
cred <- data.frame(duration=credit$duration,
                   amount=credit$amount,
                   installment_rate=credit$installment_rate,
                   present_residence=credit$present_residence,
                   age=credit$age,
                   credit_history=credit$number_credits,
                   people_liable=credit$people_liable,
                   credit_risk=as.factor(credit$credit_risk))
```

```
Error in data.frame(duration = credit$duration, amount = credit$amount, : object 'credit' not fo
und
Traceback:


1. data.frame(duration = credit$duration, amount = credit$amount,
 .     installment_rate = credit$installment_rate, present_residence = credit$present_residence,
 .     age = credit$age, credit_history = credit$number_credits,
 .     people_liable = credit$people_liable, credit_risk = as.factor(credit$credit_risk))
```

C. Although all variables in **cred** except **credit_risk** are coded as numeric, the values of one of them are also **ordered factors** rather than actual numbers. In consultation with the **data description link** from the intro, write a comment identifying the **factor variable** and briefly **describe** each variable in the dataframe.

```
head(cred,2)
```

```
##   duration amount installment_rate present_residence age credit_history
## 1        6   1169                4                 4  67              2
## 2       48   5951                2                 2  22              1
##   people_liable credit_risk
## 1             1           1
## 2             1           0
```

```
#Important Credible Variable is Credit-Risk (As.Factor)
#Duration is Month Numerical
#Amount is Credit Given
#Installment Rate is related to Credit History and Piece of Income
#Present Residence is Years at current location
#Age of Creditee (Applicant?)
#Number of Existing Credit Loans
#Number of People Liable
#Column that Represents 1/2 = 1 is good and 2 is bad?
```

Part 2: Create training and test data sets

A. Using techniques discussed in class, create **two datasets** – one for **training** and one for **testing**.

```
create_list <- createDataPartition(y = cred$credit_risk, p=2/3, list = FALSE)
training <- cred[create_list,]
testing <- cred[create_list,]
```

B. Use the dim( ) function to demonstrate that the resulting training data set and test data set contain the appropriate number of cases.

```
dim(training)
```

```
## [1] 667   8
```

```
dim(testing)
```

```
## [1] 667   8
```

```
# 8 AND 8 BOTH MATCH
```

Part 3: Build a Model using SVM

A. Using the caret package, build a support vector model using all of the variables to predict **credit_risk**

```
svm_model <- train(credit_risk ~ ., data=training, method = "svmRadial", preProc = c("center","s
cale"))
```

B. output the model

Hint: explore finalModel in the model that would created in F.

```
svm_model
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 667 samples
##   7 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (7), scaled (7)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 667, 667, 667, 667, 667, 667, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.7020770  0.04252937
##   0.50  0.7073287  0.10461350
##   1.00  0.7078732  0.13700898
##
## Tuning parameter 'sigma' was held constant at a value of 0.1410198
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.1410198 and C = 1.
```

Part 4: Predict Values in the Test Data and Create a Confusion Matrix

A. Use the predict( ) function to validate the model against test data. Store the predictions in a variable named **svmPred**.

```
svmPred <- predict(svm_model,newdata=testing)
```

B. The **svmPred** object contains a list of classifications for reliable (=0) or risky (=1) borrowers. Review the contents of **svmPred** using head( ).

```
head(svmPred)
```

```
## [1] 1 0 1 1 1 1
## Levels: 0 1
```

C. Explore the **confusion matrix**, using the caret package

```
matrix <- table(data=svmPred,testing$credit_risk)
matrix
```

```
##
## data   0   1
##    0  50  13
##    1 150 454
```

```
table(matrix)
```

```
## matrix
##  13  50 150 454
##   1   1   1   1
```

D. What is the **accuracy** based on what you see in the confusion matrix.

```
#Looking at the Table Above - 0 = 0.30 & 1 = 0.70
#30% chance of Error and 70% Chance of Accuracy
```

E. Compare your calculations with the **confusionMatrix()** function from the **caret** package.

```
matrix <- confusionMatrix(data=svmPred,reference = testing$credit_risk)
matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  50  13
##          1 150 454
##
##                Accuracy : 0.7556
##                  95% CI : (0.7212, 0.7878)
##     No Information Rate : 0.7001
##     P-Value [Acc > NIR] : 0.0008489
##
##                   Kappa : 0.2763
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.25000
##             Specificity : 0.97216
##          Pos Pred Value : 0.79365
##          Neg Pred Value : 0.75166
##              Prevalence : 0.29985
##          Detection Rate : 0.07496
##    Detection Prevalence : 0.09445
##       Balanced Accuracy : 0.61108
##
##        'Positive' Class : 0
##
```

```
#Accuracy is the Same. Different Ways of returning the predicted results. I find the confusionMa
trix() to be the best way to find an accurate result.
```

F. Explain, in a block comment:
  1) why it is valuable to have a “test” dataset that is separate from a “training” dataset, and
  2) what potential ethical challenges this type of automated classification may pose.

```
#1) 2 Different data sets allow for issues to be resolved/found easier, and create an effective
 model. 2 Different data sets allow for other options that may reduce redundancy and can help pr
ovide a better evaluation of the data included in the set.
#2)Automated Classification can cause issues with how we "Classify" people. Should we allow auto
mation for credit application? I think it also begs into question about if someone gets denied,
 is it because a computer rejection or a bias coded into the program? Lot's of ethical issues re
lated automation?
```

Part 5: Now build a tree model (with rpart)

A. Build a model with rpart
Note: you might need to install the e1071 package

```
library(rpart.plot)
```

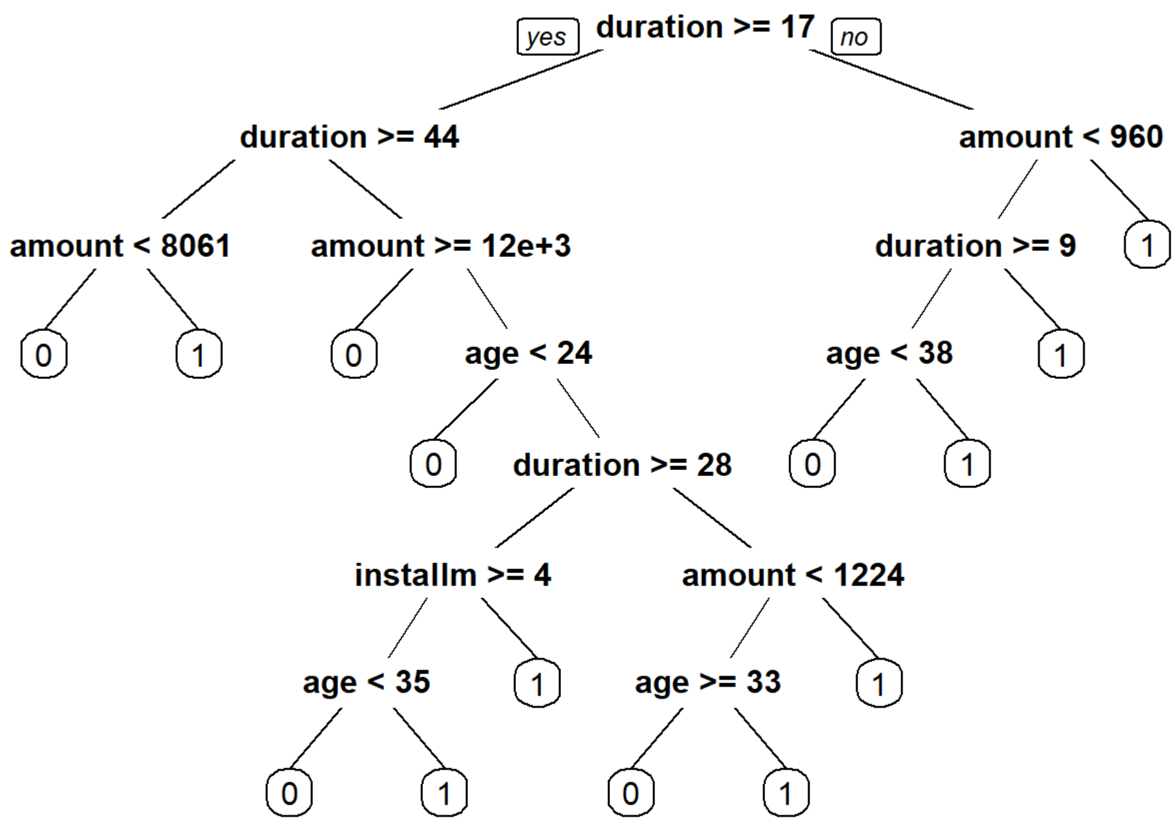
```
## Loading required package: rpart
```

```
library(rpart)
```

```
tree_model <- train(credit_risk ~ ., method = "treebag", data = training, preProc = c("center",
"scale"))
```

B. Visualize the results using rpart.plot()

```
tree_tree <- rpart(credit_risk ~ ., data = training, method = "class")
prp(tree_tree)
```




```
#Hopefully, I did this right. Getting the tree map to behave is quite diffuclt, I could also inc
lude other attritubtes to the map if need be in the comment section?
```

C. Use the **predict()** function to predict the testData, and then generate a confusion matrix to explore the results

```
predict_tree <- predict(tree_model,testing)

tree_conf_matrix <- confusionMatrix(data=predict_tree,reference = testing$credit_risk)
tree_conf_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 197   2
##          1   3 465
##
##                Accuracy : 0.9925
##                  95% CI : (0.9826, 0.9976)
##     No Information Rate : 0.7001
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9821
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9850
##             Specificity : 0.9957
##          Pos Pred Value : 0.9899
##          Neg Pred Value : 0.9936
##              Prevalence : 0.2999
##          Detection Rate : 0.2954
##    Detection Prevalence : 0.2984
##       Balanced Accuracy : 0.9904
##
##        'Positive' Class : 0
##
```

D. Review the attributes being used for this credit decision. Are there any that might not be appropriate, with respect to fairness? If so, which attribute, and how would you address this fairness situation. Answer in a comment block below

```
# I think overall from looking at the Data, the Tree Map and this last prediction confusion matr
ix, it shows that with higher age the Bank is more likely to lend credit to individuals. Younger
age requires more selections to get credit such as Job, Place Living and Income. I think to addr
ess fairness we need to rely on other variables such as past debt, income, how long at current a
ddress and credit score.
```