Intro to Data Science HW 8


```
# Enter your name here:  Ryan Tervo
# Course Number:         IST 687
# Assignment Name:       Homework #8
# Due Date:              05 Dec 2022
# Received Extension:
# Submitted Date:        12 Dec 2022
```

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

```
# 1. I did this homework by myself, with help from the book and the professor.
```

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

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

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

```
#install.packages('ggcorrplot')
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(stringr)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggcorrplot)
library(rpart)
library(rpart.plot)
library(e1071)


#   DEFINE VARIABLES:
fileName <- "https://intro-datascience.s3.us-east-2.amazonaws.com/GermanCredit.csv"

#   READ IN DATA:
credit   <- data.frame(read_csv(fileName, show_col_types = FALSE))

#   LOOK AT PORTION OF DATA:
head(credit)
```

```
##                 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
## 4        ... < 100 DM       42 existing credits paid back duly till now
## 5        ... < 100 DM       24          delay in paying off in the past
```

```
## 6 no checking account       36 existing credits paid back duly till now
##              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
## 4    radio/television   7882               ... < 100 DM  4 <= ... < 7 years
## 5           car (new)   4870               ... < 100 DM  1 <= ... < 4 years
## 6          retraining   9055 unknown/no savings account  1 <= ... < 4 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
## 4                2                      male : single     guarantor
## 5                3                      male : single          none
## 6                2                      male : single          none
##   present_residence                                     property age
## 1                 4                                  real estate  67
## 2                 2                                  real estate  22
## 3                 3                                  real estate  49
## 4                 4 building society savings agreement/life insurance  45
## 5                 4                            unknown/no property  53
## 6                 4                            unknown/no property  35
##   other_installment_plans  housing number_credits                      job
## 1                    none      own              2 skilled employee/official
## 2                    none      own              1 skilled employee/official
## 3                    none      own              1      unskilled - resident
## 4                    none for free              1 skilled employee/official
## 5                    none for free              2 skilled employee/official
## 6                    none for free              1      unskilled - resident
##   people_liable telephone foreign_worker credit_risk
## 1             1       yes            yes           1
## 2             1        no            yes           0
## 3             2        no            yes           1
## 4             2        no            yes           1
## 5             2        no            yes           0
## 6             2       yes            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))
cred$credit_risk <- as.factor(cred$credit_risk)
#   The variable we are trying to predict is '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.

```
#  Both 'Present employment since' and 'Job' appears to be an ordered factor.


#  Describe each variable in the dataframe:
#============================================================
#A. status                    |   General category of (... < 100 DM  /  0 <= ... < 200 DM  /  n
o checking account  /  ... >= 200 DM / salary for at least 1 year)
#B. duration                  |   Range of 4 to 72
#C. credit_history            |   General category of (critical account/other credits existing
  /   existing credits paid back duly till now   /   delay in paying off in the past   /   no
credits taken/all credits paid back duly   /   all credits at this bank paid back duly
#C. purpose                   |   General category of (domestic appliances   /   retraining  /
 radio/television    /   car (new)   /   car (used)   /   others   /   repairs   /   education
    /   furniture/equipment   /   business)
#E. amount                    |   Range 250 - 1824
#F. savings                   |   General category of (unknown/no savings account   /   ... < 1
00 DM   /   ... >= 1000 DM   /   500 <= ... < 1000 DM   /   100 <= ... < 500 DM)
#G. employment_duration       |   General category of (1 <= ... < 4 years   /   unemployed   ..
. >= 7 years   /   ... < 1 year   /   4 <= ... < 7 years)
#H. installment_rate          |   Range 1 to 4
#I. personal_status_sex       |   General category of (Male/Female) & (single/separated/married
/divorced status)
#J. other_debtors             |   General category of (none    /   guarantor   /   co-applicant
)
#K. present_residence         |   Ranges from  1 to 4
#L. property                  |   General category (unknown/no property  /  real estate  /  bui
lding society savings agreement/life insurance  /  car or other  /  building / car / realestat
e / unknown or no property)
#M. age                       |   Range 19 - 75
#N. other_installment_plans   |   General category of (none   /   bank   /   stores   Bank / No
ne / Stores)
#O. housing                   |   General category (For Free / Own / Rent)
#P. number_credits            |   Range between 1 to 4
#Q. job                       |   General category of (skilled employee/official   /   unemploy
ed/unskilled - non-resident   /   management/self-employed/highly qualified employee/officer
 /   unskilled - resident)
#R. people_liable             |   Range 1 to 2
#S. telephone                 |   Category/Binary (yes/no)
#T. foreign_worker            |   Category/Binary (yes/no)
#U. credit_risk               |   Category/binary (1/0)
```

Part 2: Create training and test data sets

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

```
#   SPLIT DATA:
trainingIndx <- createDataPartition(y=cred$credit_risk, p=0.80, list=FALSE)
training     <- cred[ trainingIndx, ]
testing      <- cred[-trainingIndx, ]
```

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

```
#   DEFINE VARIABLES:
dimCred    <- dim(cred)
dimTrain <- dim(training)
dimTest  <- dim(testing)

#   DEFINE PRINT STATEMENT:
printString1 <- paste('The Dimensions of Cred  are: ', dimCred[1],  ' by ', dimCred[2],    sep
 = '')
printString2 <- paste('The Dimensions of Train are: ', dimTrain[1], ' by ', dimTrain[2], sep =
 '')
printString3 <- paste('The Dimensions of Test  are: ', dimTest[1],  ' by ', dimTest[2],  sep =
 '')

#   DISPLAY RESULTS:
print(printString1, quote = FALSE)
```

```
## [1] The Dimensions of Cred  are: 1000 by 8
```

```
print(printString2, quote = FALSE)
```

```
## [1] The Dimensions of Train are: 800 by 8
```

```
print(printString3, quote = FALSE)
```

```
## [1] The Dimensions of Test  are: 200 by 8
```

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**

```
#   BUILD THE MODEL:
svmModel <- train(credit_risk ~., data = training, method = "svmRadial", preProc = c("center",
"scale"))
```

B. output the model

Hint: explore finalModel in the model

```
#   EXPLORE THE FINAL MODEL IN MODEL:
svmModel$finalModel
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 0.25
```

```
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.142411423497508
##
## Number of Support Vectors : 523
##
## Objective Function Value : -117.8907
## Training error : 0.3
```

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**.

```
#   USE 'predict()' TO EVALUATE THE MODEL AGAINST REAL TEST DATA:
svmPred <- predict(svmModel, 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().

```
#   REVIEW SVMPRED USING 'head()' FUNCTION:
head(svmPred)
```

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

C. Calculate a confusion matrix using the table function.

```
#   DEFINE CONFUSION MATRIX
confusion_matrix <- table(svmPred, testing$credit_risk)
confusion_matrix
```

```
##
## svmPred   0   1
##       0   0   0
##       1  60 140
```

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

The diag( ) command can be applied to the results of the table command you ran in the previous step. You can also use sum( ) to get the total of all four cells. Error rate: sum of diagonal vs total table

```
#   CALCULATE ACCURACY USING
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
accuracy
```

```
## [1] 0.7
```

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

```
#   CALCULATE CONFUSION MATRIX USING 'confusionMatrix()' FROM 'caret' PACKAGE:
confusion_matrix1  <- confusionMatrix(svmPred,testing$credit_risk)
confusion_matrix1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0   0   0
##          1  60 140
##
##                Accuracy : 0.7
##                  95% CI : (0.6314, 0.7626)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.5348
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 2.599e-14
##
##             Sensitivity : 0.0
##             Specificity : 1.0
##          Pos Pred Value : NaN
##          Neg Pred Value : 0.7
##              Prevalence : 0.3
##          Detection Rate : 0.0
##    Detection Prevalence : 0.0
##       Balanced Accuracy : 0.5
##
##        'Positive' Class : 0
##
```

```
print('The calculated accuracy in part D. and the confusion matrix from the caret package the same.')
```

```
## [1] "The calculated accuracy in part D. and the confusion matrix from the caret package the
 same."
```

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.

```
# When training a model we need data with the right answers.  Once the model is trained we nee
d to use data to have the model make predictions.  Then we can compare the models predictions
to the actual answers.
#
# If we use the training data to test the model it is a little bit like cheating.  We literall
y gave the model all of the questions and answers and said can you give me back what I gave yo
u.  The answer is "of course!"
```

```
#
# The solution is splitting the data into a training and test set.  This way once we have trai
ned the model then we can use the test set to make predictions and can determine if the predic
tions are accurate since we know the right answers.
#
# When looking at data in the aggregate, provided there is enough variation, then individuals
privacy is protected.  When drilling down into specific cases or records then privacy evaporat
es.  A good example is considering medical data.  If someone looks at the medical data for an
entire state and gets general statistics of the rate of a disease or condition then privacy is
 generally protected.  However, once someone starts looking into false positives and false neg
atives names then specific individuals could be identified and known.  Knowing how much detail
s and what details one is allowed to know is important to ensure privacy and legal protections
 are maintained.
```

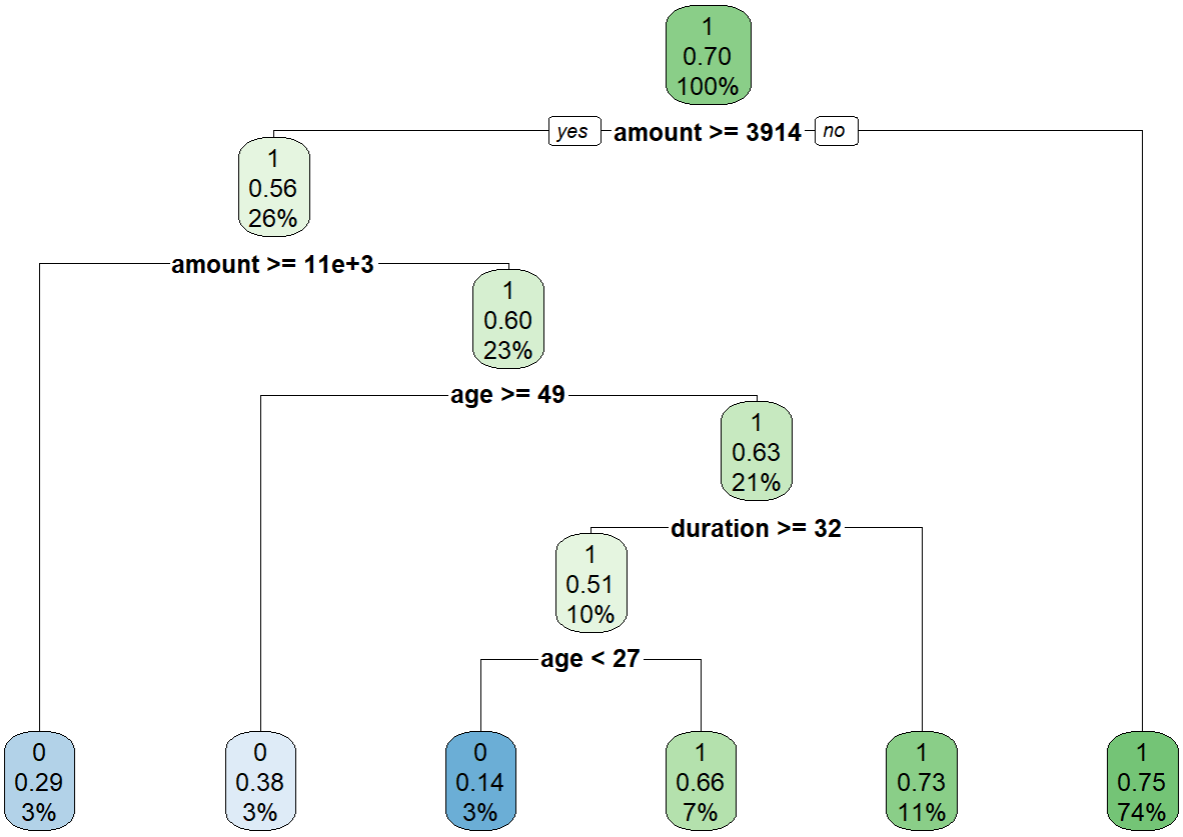
Part 5: Now build a tree model (with rpart)

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

```
#   BUILD MODEL WITH RPART:
model_tree  <- rpart(credit_risk ~ ., data = training)
```

B. Visualize the results using rpart.plot()

```
#   CREATE VISUAL:
rpart.plot(model_tree)
```



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

```
#   GET PREDICTIONS:
pred_tree <- predict(model_tree, newdata = testing, type ='class')#, type = "raw")

#   DEFINE CONFUSION MATRIX
confusion_matrix_tree <- table(pred_tree, testing$credit_risk)
confusion_matrix_tree
```

```
##
## pred_tree   0   1
##         0   8   6
##         1  52 134
```

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

```
#  It might be unfair using personal status and sex determine credit worthiness.
#-------------------------------------------------------------------------------
#  Step 1:  Try the model with and without that feature.
  # If the elimination of that feature has minimal impact on the model then just remove it.
  # If it has a significant impact go to step 2:
    #  Step 2:  Try to split the feature into two features and eliminate one at a time to see
the impact.
     ## If one of them can be removed then remove it.
```