Supervised learning for firm dynamics

Falco J. Bargagli-Stoffi, Jan Niederreiter

15/2/2020

Introduction

R Markdown

This is an `R Markdown` document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using `R Markdown` see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded `R` code chunks within the document. You can embed an R code chunks like the following.

Packages Upload

```
rm(list=ls())          # to clean the memeory
library(rpart)         # package for decision tree
library(randomForest)  # package for random forest
library(e1071)         # package for support vector machine
library(neuralnet)     # package for neural network
library(PRROC)         # package for ROC curves
```

Data Upload

In the following chuck of code you need to set the `R` working directory. Set this directory to be the path to the same folder where you will store the \texttt{mock_data.Rdata} file. Upload the data by using the `load` function.

```
setwd("G:\\Il mio Drive\\Research\\Book Chapter in Data Science for Economics and Finance\\Draft\\super
load("mock_data.Rdata")
```

A Simple Supervised Learning Routing

This simple step-by-step guide should aid the reader in designing a supervised learning (SL) routine to predict outcomes from input data.

1. Check that information on the outcome of interest is contained for the observations that are later used to train and test the SL algorithm, i.e. that the dataset is labeled. The outcome variable is the `failure` variable.

```
summary(mock_data)
```

```
##    consdummy      capital_intensity     failure      labour_product
##  Min.   :0.0000   Min.   :       0   Min.   :0.000   Min.   :-16173700
##  1st Qu.:0.0000   1st Qu.:    6632   1st Qu.:0.000   1st Qu.:    25790
##  Median :0.0000   Median :   23999   Median :1.000   Median :    42041
```

```
##  Mean   :0.0114   Mean   :   80423   Mean   :0.503   Mean   :    49644
##  3rd Qu.:0.0000   3rd Qu.:   74228   3rd Qu.:1.000   3rd Qu.:    64934
##  Max.   :1.0000   Max.   :11209132   Max.   :1.000   Max.   :  2888788
##     fin_cons             inv             ICR_failure        NEG_VA
##  Min.   :0.000001   Min.   :-79949336   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.032399   1st Qu.:      481   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.159665   Median :    14458   Median :0.000   Median :0.0000
##  Mean   :0.358747   Mean   :   403497   Mean   :0.324   Mean   :0.0374
##  3rd Qu.:0.762041   3rd Qu.:   108758   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :1.000000   Max.   :191012992   Max.   :1.000   Max.   :1.0000
##     real_SA           Z_score         misallocated_fixed profitability
##  Min.   :-17.650   Min.   :0.001156   Min.   :0.0000     Min.   :0.0000
##  1st Qu.:-11.839   1st Qu.:0.804641   1st Qu.:0.0000     1st Qu.:0.0000
##  Median :-10.723   Median :1.330870   Median :0.0000     Median :0.0000
##  Mean   :-10.853   Mean   :1.488705   Mean   :0.1694     Mean   :0.0692
##  3rd Qu.: -9.751   3rd Qu.:1.925080   3rd Qu.:0.0000     3rd Qu.:0.0000
##  Max.   : -0.990   Max.   :9.000000   Max.   :1.0000     Max.   :1.0000
##      area                zone      dummy_patents    dummy_trademark
##  Length:5000         Min.   :1.00   Min.   :0.0000   Min.   :0.000
##  Class :character    1st Qu.:3.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character    Median :3.00   Median :0.0000   Median :0.000
##                      Mean   :2.72   Mean   :0.0992   Mean   :0.103
##                      3rd Qu.:3.00   3rd Qu.:0.0000   3rd Qu.:0.000
##                      Max.   :4.00   Max.   :1.0000   Max.   :1.000
##  financial_sustainability      car             liquidity_return
##  Min.   :-0.086101        Min.   :-9731.272   Min.   :-15804.500
##  1st Qu.: 0.001978        1st Qu.:    0.072   1st Qu.:     0.001
##  Median : 0.007111        Median :    0.290   Median :     0.037
##  Mean   : 0.017525        Mean   :   -0.432   Mean   :    -3.134
##  3rd Qu.: 0.018749        3rd Qu.:    0.907   3rd Qu.:     0.083
##  Max.   : 2.512209        Max.   : 1224.616   Max.   :     2.243
##  pension_tax_debts
##  Min.   :-0.000032
##  1st Qu.: 0.000016
##  Median : 0.000035
##  Mean   : 0.001028
##  3rd Qu.: 0.000079
##  Max.   : 4.618500
```

2. Prepare the matrix of input attributes to a machine-readable format.

```
predictors <- c("consdummy", "capital_intensity", "labour_product", "fin_cons" ,
                "inv", "ICR_failure", "NEG_VA", "misallocated_fixed","profitability",
                "real_SA","Z_score", "zone","dummy_patents", "dummy_trademark",
                "financial_sustainability","car","liquidity_return","pension_tax_debts")
formula <- as.formula(paste("as.factor(failure) ~", paste(predictors, collapse="+")))
```

3. Choose how to split your data between training and testing set. Keep in mind that both training and testing set have to stay sufficiently large to train the algorithm or to validate its performance, respectively. Use resampling techniques in case of low data dimensions and stratified sampling whenever labels are highly unbalanced (undersampling, oversampling). If the data has a time dimension, make sure that the training set is formed by observations that occured before the ones in the testing set.

In the following we depict a simple 2 folds split between traning and testing data. 75% of the data will be used to train the model and 25% to test the model. Set a seed for reproducible results.

```r
set.seed(2020)
index <- sample(nrow(mock_data), size = nrow(mock_data)*0.75, replace = FALSE)
train <- mock_data[index,]
test <- mock_data[-index,]
```

4. Choose the SL algorithm that best suits your need. Possible dimensions to evaluate are prediction performance, simplicity of result interpretation and CPU runtime. Often a *horserace* between many algorithms is performed and the one with the highest prediction performance is chosen. Here, we will train all the four algorithms that we focus on: *decision tree, random forest, support vector machine* and *artificial neural network*.

5. Train the algorithms using the training set only.

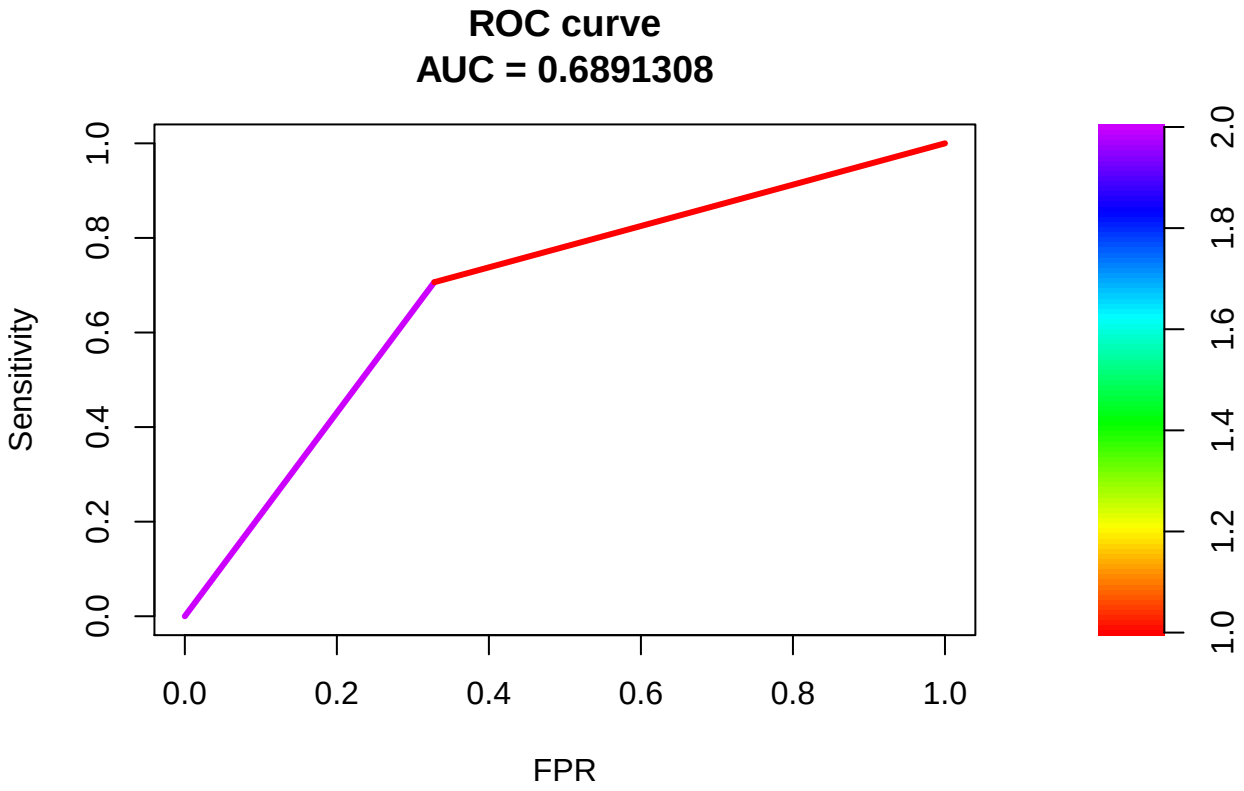
```r
set.seed(2020)
# Decision Tree
dt <- rpart(formula, data=train)

# Random Forest
rf <- randomForest(formula, data=train)

# Support Vector Machine
svm.model <- svm(formula, data=train)

# Artificial Neural Network
nnet <- neuralnet(formula, data=train)
```

6. Once the algorithms are trained, use them to predict the outcome on the testing set. Compare the predicted outcomes with the true outcomes.

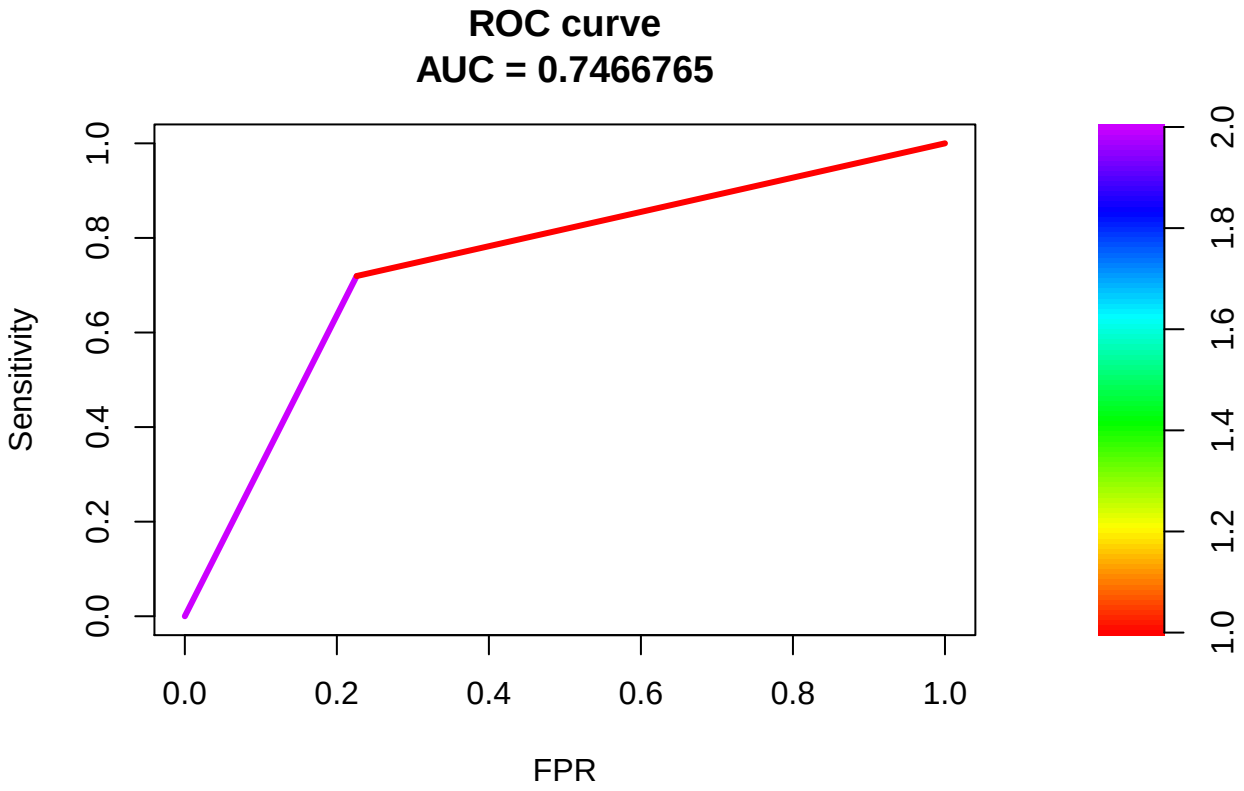
```r
# Predicted outcomes Decision Tree
dt.pred <- predict(dt, newdata=test, type='class')

# Predicted outcomes Random Forest
rf.pred <- predict(rf, newdata=test, type='class')

# Predicted outcomes Support Vector Machine
svm.pred <- predict(svm.model, newdata = test)

# Predicted Outcomes Artificial Neural Network
nnet.prob <- compute(nnet,test)
nnet.pred <- ifelse(nnet.prob$net.result[,1] < 0.5, 1, 0)
```

7. Choose the performance measure on which to evaluate the algorithms. A popular performance measure is the Area Under the receiver operating Curve (AUC).

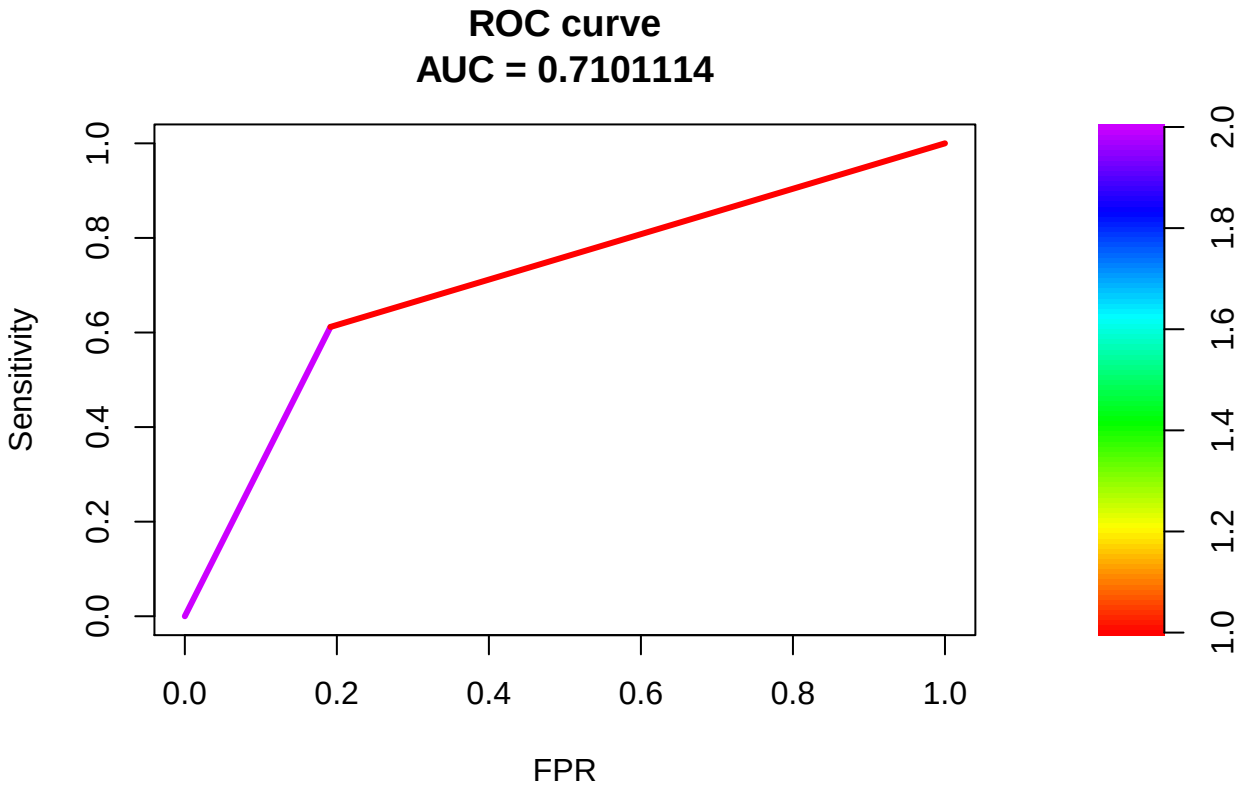
```r
# AUC Decision Tree
fg.dt <- dt.pred[test$failure==1]
bg.dt <- dt.pred[test$failure==0]
roc.dt <- roc.curve(scores.class0 = fg.dt, scores.class1 = bg.dt, curve = T)
plot(roc.dt)
```



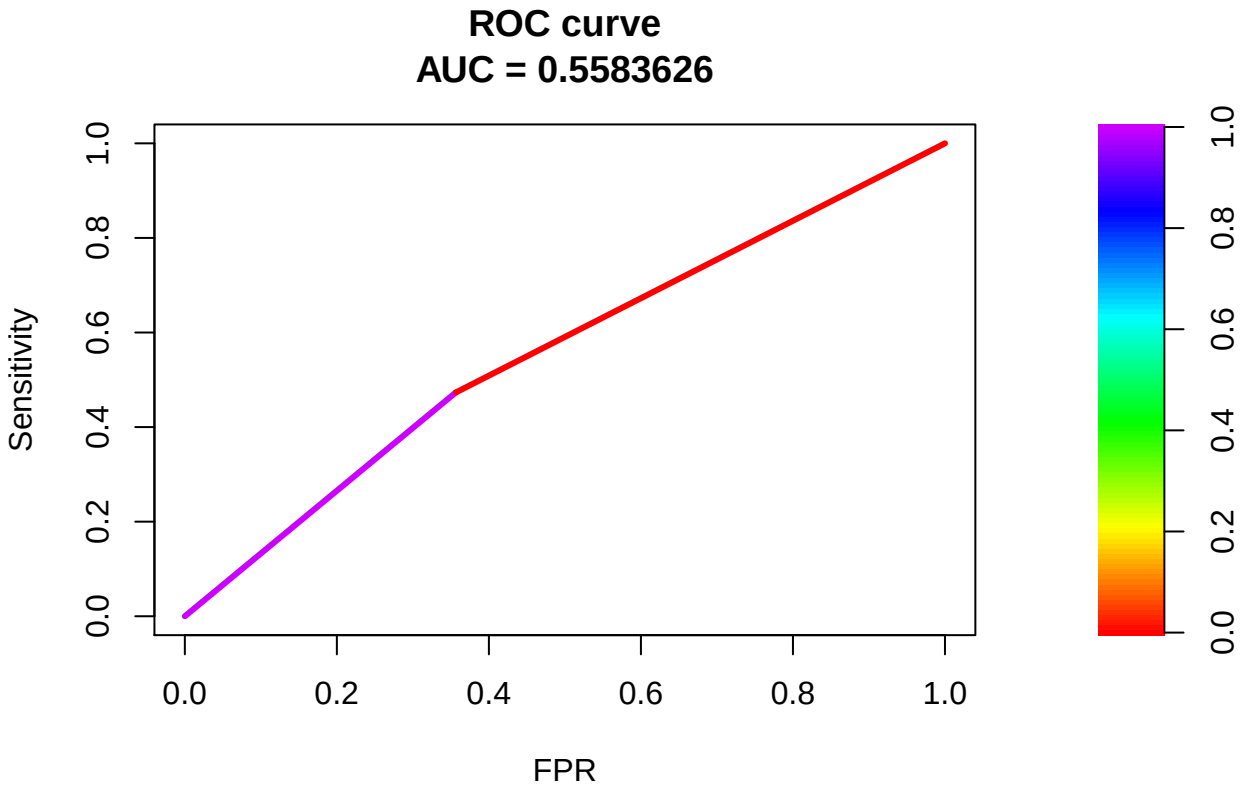

```
# AUC Random Forest
fg.rf <- rf.pred[test$failure==1]
bg.rf <- rf.pred[test$failure==0]
roc.rf <- roc.curve(scores.class0 = fg.rf, scores.class1 = bg.rf, curve = T)
plot(roc.rf)
```




```
# AUC Support Vector Machine
fg.svm <- svm.pred[test$failure==1]
bg.svm <- svm.pred[test$failure==0]
roc.svm <- roc.curve(scores.class0 = fg.svm, scores.class1 = bg.svm, curve = T)
plot(roc.svm)
```




```
# AUC Artificial Neural Network
fg.nnet <- nnet.pred[test$failure==1]
bg.nnet <- nnet.pred[test$failure==0]
roc.nnet <- roc.curve(scores.class0 = fg.nnet, scores.class1 = bg.nnet, curve = T)
plot(roc.nnet)
```



8. Once prediction performance has been assessed, the algorithm can be used to predict outcomes for observations for which the outcome is unknown. Note that valid predictions require that new observations should contain similar features and need to be independent from the outcome of old ones.