Categories

1. [Programming](https://datascienceplus.com/category/programming/)

Tags

1. [Data Visualisation](https://datascienceplus.com/tag/data-visualisation/)
2. [Flexdashboard](https://datascienceplus.com/tag/flexdashboard/)
3. [R Programming](https://datascienceplus.com/tag/rstats/)
4. [RMarkdown](https://datascienceplus.com/tag/rmarkdown/)

In this article, you learn how to make Automated Dashboard for Credit Modelling with Decision trees and Random forests in R. First you need to install the `rmarkdown` package into your R library. Assuming that you installed the `rmarkdown`, next you create a new `rmarkdown` script in R.

After this you type the following code in order to create a dashboard with rmarkdown and flexdashboard:

---

title: "Automated Dashboard for Credit Modelling with Decision trees and Random forests in R"

author: "Kristian Larsen"

output:

flexdashboard::flex\_dashboard:

orientation: rows

vertical\_layout: scroll

---

```{r setup, include=FALSE}

# Data management packages

library(flexdashboard)

library(dplyr)

library(caret)

library(partykit)

library(randomForest)

library(Hmisc)

knitr::opts\_chunk$set(cache=TRUE)

options(scipen = 9999)

rm(list=ls())

# Read dataset

loans <- read.csv("http://www.sci.csueastbay.edu/~esuess/classes/Statistics\_6620/Presentations/ml7/credit.csv")

str(loans)

# Data management

# Change the order/level of checking\_balance variable

loans$checking\_balance <- factor(loans$checking\_balance,

levels = c(" 200 DM",

"unknown"))

summary(loans[loans$default == "yes", "checking\_balance"])

# Change the order/level of saving\_balance variable

loans$savings\_balance <- factor(loans$savings\_balance,

levels = c(" 1000 DM",

"unknown"))

summary(loans[loans$default == "yes", "savings\_balance"])

# Change the order/level of credit\_history variable

loans$credit\_history <- factor(loans$credit\_history,

levels = c("critical",

"poor",

"good",

"very good",

"perfect"))

summary(loans[loans$default == "yes", "credit\_history"])

# Change the order/level of other\_credit variable

loans$other\_credit <- factor(loans$other\_credit,

levels = c("none",

"store",

"bank"))

summary(loans[loans$default == "yes", "other\_credit"])

set.seed(300)

in\_loans\_train <- sample(nrow(loans), nrow(loans)\*0.75)

loans\_train <- loans[in\_loans\_train, ]

loans\_test <- loans[-in\_loans\_train, ]

```

Row {data-width=350}

-----------------------------------------------------------------------

### Chart A - Decision tree Model I

```{r}

loans\_model\_dt <- ctree(default ~ ., loans\_train)

plot(loans\_model\_dt)

```

### Chart B - Decision tree Model I - simple

```{r}

plot(loans\_model\_dt, type = "simple")

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart C - Decision tree Model Model I - formula

```{r}

loans\_model\_dt

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart D - Confusion Matrix for Decision tree Model Model I

```{r}

loans\_pred\_dt <- predict(loans\_model\_dt, loans\_test)

dt\_conft <- table("prediction" = loans\_pred\_dt,

"actual" = loans\_test$default

)

accu\_dt <- round((dt\_conft[1]+dt\_conft[4])/sum(dt\_conft[1:4]),4)

prec\_dt <- round(dt\_conft[4]/(dt\_conft[2]+dt\_conft[4]), 4)

reca\_dt <- round(dt\_conft[4]/(dt\_conft[4]+dt\_conft[3]), 4)

spec\_dt <- round(dt\_conft[1]/(dt\_conft[1]+dt\_conft[2]), 4)

confusionMatrix(loans\_pred\_dt, loans\_test$default, positive = "yes")

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart E - Decision tree Model II

```{r}

loans\_model\_dt2 <- ctree(default ~ ., loans\_train, control = ctree\_control(mincriterion = 0.7))

plot(loans\_model\_dt2)

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart F - Decision tree Model Model II - formula

```{r}

loans\_model\_dt2

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart G - Confusion Matrix for Decision tree Model Model II

```{r}

loans\_pred\_dt2 <- predict(loans\_model\_dt2, loans\_test)

confusionMatrix(loans\_pred\_dt2, loans\_test$default, positive = "yes")

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart H - Random Forest Model

```{r}

set.seed(300)

ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3, allowParallel = TRUE)

loans\_rf <- train(default ~ ., data = loans, method = "rf", trControl = ctrl)

loans\_rf$finalModel

```

Row {data-width=650}

-----------------------------------------------------------------------

### Chart I - Random Forest Model - variable importance

```{r}

varImp(loans\_rf)

```

Row {data-width=350}

-----------------------------------------------------------------------

### Chart J - Random Forest Model - Final model plot I

```{r}

plot(loans\_rf$finalModel)

legend("topright", colnames(loans\_rf$finalModel$err.rate),col = 1:6, cex = 0.8, fill = 1:6)

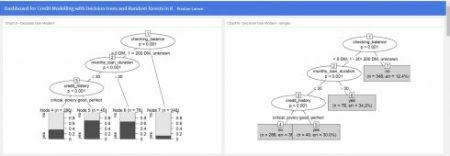
```

### Chart K - Random Forest Model - Final model plot II

```{r}

plot(loans\_rf)

```

Screenshot:  
[](https://i0.wp.com/datascienceplus.com/wp-content/uploads/2019/01/Dashboard-for-Credit-Modelling-with-Decision-trees-and-Random-forests-in-R.jpg?ssl=1)

The result of the above coding are published with RPubs [here](http://rpubs.com/knl84/457080).

**References**

1. [Using flexdashboard in R](https://rmarkdown.rstudio.com/flexdashboard/)