```
##################################################################################
# Project:      Credit Scoring Analysis (petit example)
# Description:  Part 7 - Logistic Regression
# Data:         CleanCreditScoring.csv
# By:           Gaston Sanchez
# url:          www.gastonsanchez.com
#
# Apply logistic regression on the results obtained
##################################################################################

# remember to change your working directory!!! (don't use mine)
# setwd("/Users/gaston/Documents/Gaston/StatsDataMining")

# load package FactoMineR and ggplot2
require(FactoMineR)
require(ggplot2)

# read cleaned data set
dd = read.csv("CleanCreditScoring.csv", header=TRUE, stringsAsFactors=TRUE)


# ================================================================================
# Apply logistic regression using all the variables
# ================================================================================

# let's keep 2/3 of the data for learning, and 1/3 for testing
n = nrow(dd)
learn = sample(1:n, size=round(0.67 * n))
nlearn = length(learn)
ntest = n - nlearn

# "whole enchilada" logistic regression model
gl1 = glm(Status ~ ., data=dd[learn,], family=binomial)

# use the 'summary' function to obtain more details on the logistic model
# (pay attention to the signicance of the coefficients)
summary(gl1)

# let's use the 'anova' function to get a sequential analysis of
# variance of the model fit (ie importance of variables in the model)
anova(gl1)

# we can also use the 'step' function to perform model selection by
# applying a backward elimination method based on AIC (Akaike Info. Criterion)
step(gl1)


# ================================================================================
# Apply logistic regression after removing some variables
# ================================================================================

# new model
glf <- glm(formula = Status ~ Seniority + Age + Income + Debt + Amount + Finrat +
  seniorityR + expensesR + assetsR + priceR + savingsR + Home + Marital + Records +
  Job, family = binomial, data = dd[learn, ])
# check summary
summary(glf)
# check coefficients
exp(glf$coefficients)

# re-expressed fitted values
glf$fitted.values = 1 - glf$fitted.values

# create vector for predictions
```

```r
glfpred = rep(NA, length(glf$fitted.values))
glfpred[glf$fitted.values < 0.5] = 0
glfpred[glf$fitted.values >= 0.5] = 1

# how is the prediction? (confusion matrix)
table(dd$Status[learn], glfpred)
# error rate
error_rate.learn = 100*sum(diag(table(dd$Status[learn], glfpred))) / nlearn
error_rate.learn

# let's use the test data to get predictions
glft = predict(glf, newdata=dd[-learn,])
pt = 1 / (1 + exp(-glft))
pt = 1 - pt

# vector of predicted values
glftpred = rep(NA, length(pt))
glftpred[pt < 0.5] = 0
glftpred[pt >= 0.5] = 1

# confusion matrix
table(dd$Status[-learn], glftpred)
error_rate.test = 100*sum(diag(table(dd$Status[-learn], glftpred))) / ntest
error_rate.test


# ==================================================================================
# Concentration curve
# ==================================================================================

ac_tot = 100*(1:ntest) / ntest

pt.ord = order(pt, decreasing=T)

Status_test = dd$Status[-learn]
npos = table(Status_test)[2]

ac_pos_test = 100*cumsum(Status_test[pt.ord] == "good") / npos

plot(ac_tot, ac_pos_test, type="l", lwd=2,
     main="Concentration Curve")
lines(ac_tot, ac_tot, col="gray70", lwd=2)

# ==================================================================================
# ROC Curve
# ==================================================================================

nneg = ntest - npos

ac_neg_test = 100*cumsum(Status_test[pt.ord]=="bad") / nneg

plot(ac_neg_test, ac_pos_test, type="l", lwd=2, main="ROC Curve", col="blue")
lines(ac_neg_test, ac_neg_test, col="grey70", lwd=2)
```