Machine Learning Script for the Prediction of Stability Score of a country

Adrien Ratsimbaharison

July 16, 2019

## Introduction

In this machine learning implementation, we follow the guidelines suggested by different data scientists who specialize in the use r statistical and programming language and particularly the Caret package, created and maintained by Max Kuhn. Among these guidelines, we found particularly useful Saurav Kaushik’s “Practical guide to implement machine learning with CARET in R” and Brett Lanz’s “Machine Learning with R.” After the initial step of installing the Caret package and loading the dataset into r, this machine learning implementation includes the following:

* defining the problem,
* preprocessing the data,
* spliting the data into training and test sets,
* feature selection using the “recursive feature elimination”" or “rfe”" function,
* traning models on the training set,
* generating variable importance,
* making predictions on the test set and assessing the accuracy of the predictions.

## 1. Getting started with loading the package, looking at the data, and defining the problem

Installing and loading the Caret package and its dependencies:

Reading the data in R and looking at its structure:

# Reading the data  
  
stabilityFullDataset <- read.csv("WGI2popDevIneqPovRegimeConflict2.csv", header = TRUE)  
  
stabilityFullDataset <- as.data.frame(stabilityFullDataset)  
  
# Selecting the variables of interest  
  
stabilityFullDataset <- select(stabilityFullDataset, date, stability, corruptionControl, governmentEffectiveness, regulatoryQuality, ruleOfLaw, voiceAndAccountability, population, GNIperCapita, GDPannualGrowthRate, HDI, GINI, povertyHeadCount, status, inverse\_pr, inverse\_cl, inverse\_mean, politicalChangeFH, conflictHistory, region, subregion)  
  
# Correcting the types of some variables  
stabilityFullDataset$date <- as.numeric(stabilityFullDataset$date)  
stabilityFullDataset$population <- as.numeric(stabilityFullDataset$population)  
stabilityFullDataset$date <- as.numeric(stabilityFullDataset$date)  
stabilityFullDataset$inverse\_pr <- as.numeric(stabilityFullDataset$inverse\_pr)  
stabilityFullDataset$inverse\_cl <- as.numeric(stabilityFullDataset$inverse\_cl)  
stabilityFullDataset$inverse\_mean <- as.numeric(stabilityFullDataset$inverse\_mean)  
stabilityFullDataset$conflictHistory <- as.factor(stabilityFullDataset$conflictHistory)  
  
# Looking at its structure of the full dataset  
str(stabilityFullDataset)

## 'data.frame': 4010 obs. of 21 variables:  
## $ date : num 1996 1998 2000 2002 2003 ...  
## $ stability : num 1.02 0.99 0.99 0.99 1.17 ...  
## $ corruptionControl : num 1.543 1.596 1.658 1.228 0.162 ...  
## $ governmentEffectiveness: num 1.77177 1.97887 2.04215 1.99657 0.00406 ...  
## $ regulatoryQuality : num 1.783 1.778 1.849 1.639 0.244 ...  
## $ ruleOfLaw : num 0.809 0.909 0.862 0.859 0.862 ...  
## $ voiceAndAccountability : num 0.227 1.006 0.945 0.946 1.115 ...  
## $ population : num 83200 87277 90853 94992 97017 ...  
## $ GNIperCapita : num 3114 3106 18286 3331 3395 ...  
## $ GDPannualGrowthRate : num 1.19 1.99 7.62 -3.27 1.98 ...  
## $ HDI : num 0.671 0.676 0.756 0.664 0.673 0.789 0.695 0.777 0.707 0.71 ...  
## $ GINI : num 53.6 42.3 37.3 45.2 40.9 42.2 44.4 38.1 37.3 45.6 ...  
## $ povertyHeadCount : num 14.1 14.6 5.9 2.8 10.7 2.5 3.1 1.5 1.3 6.7 ...  
## $ status : Factor w/ 3 levels "Free","Not Free",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ inverse\_pr : num 7 7 7 7 7 7 7 7 7 7 ...  
## $ inverse\_cl : num 7 7 7 6 7 7 7 7 7 7 ...  
## $ inverse\_mean : num 7 6.5 6.5 6.5 7 7 7 7 7 7 ...  
## $ politicalChangeFH : Factor w/ 3 levels "autocratization",..: 3 1 3 3 2 3 3 3 3 3 ...  
## $ conflictHistory : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ region : Factor w/ 5 levels "Africa","Americas",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ subregion : Factor w/ 22 levels "Australia and New Zealand",..: 2 2 2 2 2 2 2 2 2 2 ...

Defining the problem:

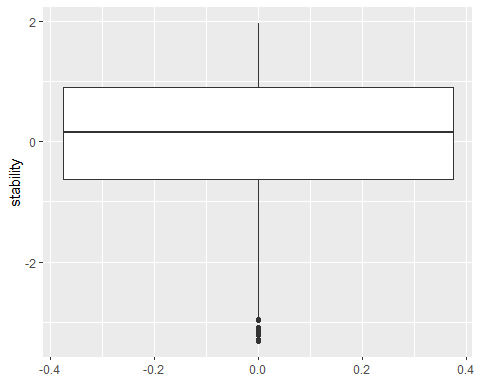
The problem in this machine learning is to predict the stability score of a country. In other words, we are dealing here with a machine learning regression on the variable “stability”.

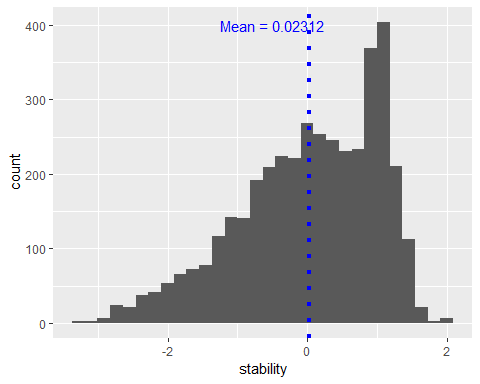
Feature engineering:

# summary statistics of the stability scores  
summary(stabilityFullDataset$stability)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.31494 -0.63062 0.15040 0.02312 0.90009 1.96506

ggplot(stabilityFullDataset, aes(y= stability)) +  
 geom\_boxplot()





As shown in these figures, the stability score around the world during the period of 1996-2017 was slightly positive and skewed to the left. This outcome variable needs to be centered and scaled to get meaningful statistical results.

## 2. Pre-processing the data using Caret

In this pre-processing step, we first check for the missing values and remove them.

# Checking for missing values  
sum(is.na(stabilityFullDataset))

## [1] 418

# Removing NAs  
stabilityFullDataset <- na.omit(stabilityFullDataset)  
sum(is.na(stabilityFullDataset))

## [1] 0

Next, we are centering and scaling the numerical values:

# centering and scaling the numerical variable educationLevel  
preProcValues <- preProcess(stabilityFullDataset, method = c("center","scale"))  
  
stabilityFullDataset\_processed <- predict(preProcValues, stabilityFullDataset)  
sum(is.na(stabilityFullDataset\_processed))

## [1] 0

Then, we create “one hot encoding” for the factor variables:

#Converting every categorical variable to numerical using dummy variables  
dmy <- dummyVars(" ~ .", data = stabilityFullDataset\_processed,fullRank = T)  
stabilityFullDataset\_processed <- data.frame(predict(dmy, newdata = stabilityFullDataset\_processed))  
  
#Checking the structure of transformed train file  
str(stabilityFullDataset\_processed)

## 'data.frame': 3801 obs. of 46 variables:  
## $ date : num -1.95 -1.616 -1.283 -0.949 -0.782 ...  
## $ stability : num 1.016 0.982 0.982 0.982 1.158 ...  
## $ corruptionControl : num 1.48 1.53 1.59 1.17 0.13 ...  
## $ governmentEffectiveness : num 1.6761 1.8772 1.9387 1.8944 -0.0402 ...  
## $ regulatoryQuality : num 1.722 1.716 1.787 1.579 0.199 ...  
## $ ruleOfLaw : num 0.797 0.896 0.85 0.846 0.85 ...  
## $ voiceAndAccountability : num 0.223 1.005 0.944 0.945 1.115 ...  
## $ population : num -0.257 -0.257 -0.257 -0.257 -0.257 ...  
## $ GNIperCapita : num -0.512 -0.512 0.367 -0.499 -0.495 ...  
## $ GDPannualGrowthRate : num -0.474 -0.333 0.65 -1.253 -0.336 ...  
## $ HDI : num -0.0473 -0.0153 0.4972 -0.0922 -0.0345 ...  
## $ GINI : num 1.8142 0.179 -0.5445 0.5987 -0.0235 ...  
## $ povertyHeadCount : num 0.0421 0.0702 -0.4187 -0.5928 -0.149 ...  
## $ status.Not.Free : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ status.Partly.Free : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ inverse\_pr : num 1.02 1.02 1.02 1.02 1.02 ...  
## $ inverse\_cl : num 1.159 1.159 1.159 0.619 1.159 ...  
## $ inverse\_mean : num 1.103 0.849 0.849 0.849 1.103 ...  
## $ politicalChangeFH.democratization: num 0 0 0 0 1 0 0 0 0 0 ...  
## $ politicalChangeFH.no.change : num 1 0 1 1 0 1 1 1 1 1 ...  
## $ conflictHistory.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ region.Americas : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ region.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ region.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ region.Oceania : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Caribbean : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ subregion.Central.America : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Central.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Eastern.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Eastern.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Eastern.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Melanesia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Micronesia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Middle.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Northern.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Northern.America : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Northern.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Polynesia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.South.Eastern.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.South.America : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Southern.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Southern.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Southern.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Western.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Western.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Western.Europe : num 0 0 0 0 0 0 0 0 0 0 ...

## 3. Splitting the data using Caret

In this step, we splitt the dataset into trainSet and testSet based on outcome with a ratio of 65% and 35%, using createDataPartition in Caret.

#Spliting dataset into trainSet and testSet  
set.seed(1234)  
index <- createDataPartition(stabilityFullDataset\_processed$stability, p=0.75, list=FALSE)  
stabilityTrainSet <- stabilityFullDataset\_processed[index,]  
stabilityTestSet <- stabilityFullDataset\_processed[-index,]  
  
#Checking the structure of approvalTrainSet  
str(stabilityTrainSet)

## 'data.frame': 2853 obs. of 46 variables:  
## $ date : num -1.95 -1.616 -1.283 -0.782 -0.448 ...  
## $ stability : num 1.016 0.982 0.982 1.158 1.37 ...  
## $ corruptionControl : num 1.48 1.53 1.59 0.13 1.22 ...  
## $ governmentEffectiveness : num 1.6761 1.8772 1.9387 -0.0402 1.2111 ...  
## $ regulatoryQuality : num 1.722 1.716 1.787 0.199 0.815 ...  
## $ ruleOfLaw : num 0.797 0.896 0.85 0.85 0.845 ...  
## $ voiceAndAccountability : num 0.223 1.005 0.944 1.115 1.153 ...  
## $ population : num -0.257 -0.257 -0.257 -0.257 -0.257 ...  
## $ GNIperCapita : num -0.512 -0.512 0.367 -0.495 -0.478 ...  
## $ GDPannualGrowthRate : num -0.474 -0.333 0.65 -0.336 -0.469 ...  
## $ HDI : num -0.0473 -0.0153 0.4972 -0.0345 0.1064 ...  
## $ GINI : num 1.8142 0.179 -0.5445 -0.0235 0.4829 ...  
## $ povertyHeadCount : num 0.0421 0.0702 -0.4187 -0.149 -0.576 ...  
## $ status.Not.Free : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ status.Partly.Free : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ inverse\_pr : num 1.02 1.02 1.02 1.02 1.02 ...  
## $ inverse\_cl : num 1.16 1.16 1.16 1.16 1.16 ...  
## $ inverse\_mean : num 1.103 0.849 0.849 1.103 1.103 ...  
## $ politicalChangeFH.democratization: num 0 0 0 1 0 0 0 0 0 0 ...  
## $ politicalChangeFH.no.change : num 1 0 1 0 1 1 1 1 1 1 ...  
## $ conflictHistory.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ region.Americas : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ region.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ region.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ region.Oceania : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Caribbean : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ subregion.Central.America : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Central.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Eastern.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Eastern.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Eastern.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Melanesia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Micronesia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Middle.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Northern.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Northern.America : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Northern.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Polynesia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.South.Eastern.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.South.America : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Southern.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Southern.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Southern.Europe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Western.Africa : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Western.Asia : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ subregion.Western.Europe : num 0 0 0 0 0 0 0 0 0 0 ...

## 4. Feature selection using Caret

In this step, we use the “recursive feature elimination” or “rfe” function in Caret to identify the best subset of features to be included in the models.

# Feature selection using rfe in caret  
ctrl <- rfeControl(functions = rfFuncs,  
 method = "repeatedcv",  
 repeats = 3,  
 verbose = FALSE)  
  
y <- stabilityTrainSet$stability  
x <- select(stabilityTrainSet, - stability)  
  
stabilityProfile <- rfe(x, y,  
 rfeControl = ctrl)  
  
stabilityProfile

##   
## Recursive feature selection  
##   
## Outer resampling method: Cross-Validated (10 fold, repeated 3 times)   
##   
## Resampling performance over subset size:  
##   
## Variables RMSE Rsquared MAE RMSESD RsquaredSD MAESD Selected  
## 4 0.4938 0.7852 0.3779 0.05132 0.03738 0.04111   
## 8 0.3124 0.9068 0.2286 0.01670 0.01079 0.01168   
## 16 0.2802 0.9248 0.2051 0.01751 0.01017 0.01336   
## 45 0.2765 0.9268 0.2021 0.01536 0.00945 0.01199 \*  
##   
## The top 5 variables (out of 45):  
## population, ruleOfLaw, conflictHistory.1, HDI, subregion.Western.Asia

## 5. Training models on training set using Caret

In this step, we train the generalized linear model (glm) on the train set:

stabilityModel\_glm <- train(stability ~ ., data = stabilityTrainSet,   
 method = "glm")

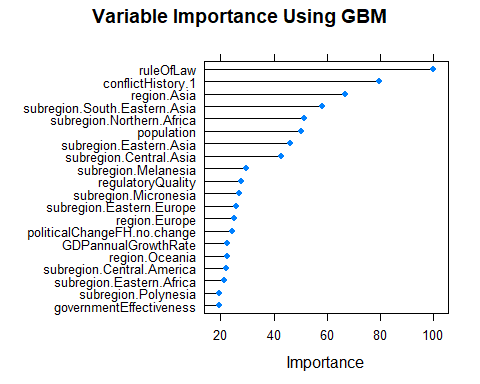
## 6. Variable importance estimation on training set using Caret

In this step, we check the variable importance estimates in Caret by using the “varImp” function" for the glm model.

# Checking variable importance with glm  
# Variable Importance  
varImp(object=stabilityModel\_glm)

## glm variable importance  
##   
## only 20 most important variables shown (out of 41)  
##   
## Overall  
## ruleOfLaw 100.00  
## conflictHistory.1 79.59  
## region.Asia 66.71  
## subregion.South.Eastern.Asia 58.33  
## subregion.Northern.Africa 51.29  
## population 50.32  
## subregion.Eastern.Asia 46.00  
## subregion.Central.Asia 42.79  
## subregion.Melanesia 29.66  
## regulatoryQuality 27.67  
## subregion.Micronesia 27.03  
## subregion.Eastern.Europe 25.72  
## region.Europe 25.07  
## politicalChangeFH.no.change 24.15  
## GDPannualGrowthRate 22.44  
## region.Oceania 22.28  
## subregion.Central.America 21.92  
## subregion.Eastern.Africa 21.27  
## subregion.Polynesia 19.44  
## governmentEffectiveness 19.32

#Plotting Variable importance for GBM model  
plot(varImp(object=stabilityModel\_glm),main="Variable Importance Using GBM", top = 20)



## 7. Making predictions on test set using Caret

#Predictions with glm  
stabilityPrediction\_glm <- predict.train(object=stabilityModel\_glm,stabilityTestSet,type="raw")  
  
summary(stabilityPrediction\_glm)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -2.10219 -0.73293 -0.09013 -0.01573 0.81387 1.90561

stabilityModel\_glm

## Generalized Linear Model   
##   
## 2853 samples  
## 45 predictor  
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 2853, 2853, 2853, 2853, 2853, 2853, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 0.5101674 0.7436278 0.3945856

# Assessing the accuracy of the prediction  
  
postResample(pred = stabilityPrediction\_glm, obs = stabilityTestSet$stability)

## RMSE Rsquared MAE   
## 0.4941748 0.7474720 0.3849288

## Conclusion

The top 5 variables (out of 45): population, ruleOfLaw, conflictHistory.1, HDI, GNIperCapita

1. Prediction of stability of the test set

glm variable importance

only 20 most important variables shown (out of 41)

Overall

ruleOfLaw 100.00 conflictHistory.1 69.71 region.Asia 57.90 subregion.South.Eastern.Asia 53.23 subregion.Northern.Africa 46.84 subregion.Central.Asia 45.97 population 42.63 subregion.Eastern.Asia 41.16 regulatoryQuality 37.49 subregion.Melanesia 26.25 region.Europe 25.49 subregion.Micronesia 23.59 subregion.Eastern.Europe 23.29 governmentEffectiveness 23.09 subregion.Central.America 21.35 region.Oceania 20.21 GDPannualGrowthRate 19.33 subregion.Eastern.Africa 18.03 politicalChangeFH.no.change 17.90 date 17.08