Script for the Machine Learning Predictions

Adrien M. Ratsimbaharison

March 17, 2019

## The six steps in the machine learning predictions:

These steps generally follow the steps suggested by different machine learning practitioners and instructors, including Lantz (2015), Pierobon (2018), and Leek, Peng, & Caffo (2018).

1. Defining the problem
2. Exploring the Data
3. Preparing the data: data randomization and partition
4. Modeling and predicting with classification
5. Modeling and predicting with regression
6. Conclusion

# Loading the required packages:  
library(caret)  
library(dplyr)  
library(ggplot2)  
library(readr)  
library(rattle)

### Step 1: Defining the problem

When performing machine learning techniques, it is always a good idea to define exactly What you are trying to predict. In our case, we are trying to predict:

1. the classification of countries either as “stable” (with stability scores greater than 0) or as “unstable” (with stability scores less than 0), and
2. the stability scores of countries based on some predictors.

In terms of machine learning, we are dealing in this case with a multivariate supervised machine learning problem in which we have to predict a binary and multi-class outcome (thus we will use classification techniques) and a numeric outcome (thus we will use a mutliple regression technique).

### Step 2: Reading and exploring the Data in r

Our data were downloaded from different sources and merged in a single data frame (See script on downloading quantitative data for the information on how the data were collected and manipulated).

We just need here to read the data into r and explore the features.

# reading the data into r  
  
WGIdevRegimeType <- read.csv("WGIdevRegimeType.csv")  
  
# looking at the different types of variables and correcting their classes  
  
WGIdevRegimeType$stabilityDummy <- as.factor(WGIdevRegimeType$stabilityDummy)  
  
glimpse(WGIdevRegimeType)

## Observations: 4,846  
## Variables: 46  
## $ X <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,...  
## $ country <fct> Aruba, Aruba, Aruba, Aruba, Aruba, Aru...  
## $ M49Code <int> 533, 533, 533, 533, 533, 533, 533, 533...  
## $ iso2c <fct> AW, AW, AW, AW, AW, AW, AW, AW, AW, AW...  
## $ iso3c <fct> ABW, ABW, ABW, ABW, ABW, ABW, ABW, ABW...  
## $ date <int> 1996, 1998, 2000, 2002, 2003, 2004, 20...  
## $ stability <dbl> 1.0233479, 0.9895858, 0.9895858, 0.989...  
## $ stabilityDummy <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## $ stabilityCategory <fct> Highly Stable, Moderately Stable, Mode...  
## $ corruptionControl <dbl> 1.5427264, 1.5964284, 1.6577593, 1.228...  
## $ governmentEffectiveness <dbl> 1.771766543, 1.978873014, 2.042146683,...  
## $ regulatoryQuality <dbl> 1.7832654, 1.7777016, 1.8493328, 1.638...  
## $ ruleOfLaw <dbl> 0.8091406, 0.9085175, 0.8622218, 0.858...  
## $ voiceAndAccountability <dbl> 0.2265435, 1.0056144, 0.9448211, 0.946...  
## $ GNIperCapita <dbl> 12044.728, 17034.389, 18399.583, 17810...  
## $ devCategory <fct> High Income, High Income, High Income,...  
## $ GDPannualGrowthRate <dbl> 1.18578999, 1.99198836, 7.61658970, -3...  
## $ HDI <dbl> 0.735, 0.775, 0.756, 0.792, 0.804, 0.8...  
## $ GINI <dbl> 55.4, 44.7, 37.3, 47.0, 35.3, 33.3, 47...  
## $ povertyHeadCount <dbl> 13.3, 5.1, 6.5, 6.6, 0.7, 0.7, 3.1, 1....  
## $ polityScore <int> 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 10...  
## $ polityCategory <fct> Democracy, Democracy, Democracy, Democ...  
## $ politicalChange <fct> no change, no change, no change, no ch...  
## $ democ <int> 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 10...  
## $ autoc <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...  
## $ durable <int> 8, 10, 10, 10, 14, 14, 14, 14, 15, 17,...  
## $ xrreg <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,...  
## $ xrcomp <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,...  
## $ xropen <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,...  
## $ xconst <int> 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7,...  
## $ parreg <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,...  
## $ parcomp <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,...  
## $ exrec <int> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,...  
## $ exconst <int> 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7,...  
## $ polcomp <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10...  
## $ pr <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## $ cl <int> 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## $ sum <int> 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2,...  
## $ mean <dbl> 1.5, 1.5, 1.5, 1.5, 1.0, 1.0, 1.0, 1.0...  
## $ status <fct> Free, Free, Free, Free, Free, Free, Fr...  
## $ inverse\_pr <int> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,...  
## $ inverse\_cl <int> 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7,...  
## $ inverse\_mean <dbl> 6.5, 6.5, 6.5, 6.5, 7.0, 7.0, 7.0, 7.0...  
## $ politicalChangeFH <fct> no change, no change, no change, no ch...  
## $ region <fct> Americas, Americas, Americas, Americas...  
## $ subregion <fct> Caribbean, Caribbean, Caribbean, Carib...

Before preparing the data for the modeling, we need to take a look at the target variables: stabilityDummy and stability.

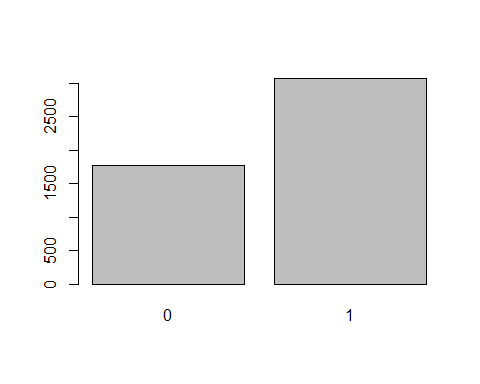
# summary statistics of the target variables  
table(WGIdevRegimeType$stabilityDummy)

##   
## 0 1   
## 1778 3068

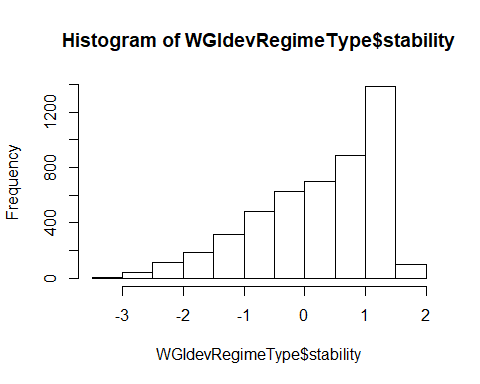
summary(WGIdevRegimeType$stability)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.3149 -0.4475 0.4544 0.2265 1.0807 1.9651

# plotting the target variables  
  
barplot(table(WGIdevRegimeType$stabilityDummy))



hist(WGIdevRegimeType$stability)



We need also to take a look at the summary statistics of the other variables of interest, paying attention to the existence of NAs, outliers, and zero variances.

WGIdevRegimeTypeSummary <- select(WGIdevRegimeType, stability, stabilityDummy, stabilityCategory, corruptionControl, governmentEffectiveness, regulatoryQuality, ruleOfLaw, voiceAndAccountability, GNIperCapita, devCategory, GDPannualGrowthRate, HDI, GINI, povertyHeadCount, polityScore, polityCategory, politicalChange, democ, autoc, durable, xrreg, xrcomp, xropen, xconst, parreg, parcomp, exrec, exconst, polcomp, pr, cl, sum, mean, status, inverse\_pr, inverse\_cl, inverse\_mean, politicalChangeFH, region, subregion)  
  
summary(WGIdevRegimeTypeSummary)

## stability stabilityDummy stabilityCategory  
## Min. :-3.3149 0:1778 Highly Stable :1498   
## 1st Qu.:-0.4475 1:3068 Highly Unstable : 667   
## Median : 0.4544 Moderately Stable :1570   
## Mean : 0.2265 Moderately Unstable:1111   
## 3rd Qu.: 1.0807   
## Max. : 1.9651   
##   
## corruptionControl governmentEffectiveness regulatoryQuality  
## Min. :-1.8687 Min. :-2.4784 Min. :-2.6450   
## 1st Qu.:-0.6352 1st Qu.:-0.6125 1st Qu.:-0.5432   
## Median : 0.1665 Median : 0.2024 Median : 0.2839   
## Mean : 0.2928 Mean : 0.3317 Mean : 0.2879   
## 3rd Qu.: 1.3334 3rd Qu.: 1.4710 3rd Qu.: 1.3565   
## Max. : 2.4700 Max. : 2.4370 Max. : 2.2605   
##   
## ruleOfLaw voiceAndAccountability GNIperCapita   
## Min. :-2.6064 Min. :-2.3134 Min. : 204.9   
## 1st Qu.:-0.6593 1st Qu.:-0.5750 1st Qu.: 1295.9   
## Median : 0.2753 Median : 0.4306 Median : 4000.5   
## Mean : 0.2204 Mean : 0.2189 Mean : 11621.9   
## 3rd Qu.: 1.0816 3rd Qu.: 1.1133 3rd Qu.: 17276.4   
## Max. : 2.1003 Max. : 1.8010 Max. :119043.8   
##   
## devCategory GDPannualGrowthRate HDI   
## High Income :1452 Min. :-62.076 Min. :0.2350   
## Low Income :1007 1st Qu.: 1.642 1st Qu.:0.5693   
## Lower Middle Income:1439 Median : 3.770 Median :0.6960   
## Upper Middle Income: 948 Mean : 3.856 Mean :0.6746   
## 3rd Qu.: 5.978 3rd Qu.:0.7930   
## Max. :149.973 Max. :0.9530   
##   
## GINI povertyHeadCount polityScore polityCategory  
## Min. :16.20 Min. : 0.0 Min. :-10.000 Anocracy :1159   
## 1st Qu.:37.30 1st Qu.: 1.3 1st Qu.: 5.000 Autocracy: 468   
## Median :42.10 Median : 5.8 Median : 9.000 Democracy:3219   
## Mean :41.21 Mean :14.4 Mean : 5.869   
## 3rd Qu.:45.00 3rd Qu.:21.0 3rd Qu.: 10.000   
## Max. :65.80 Max. :94.1 Max. : 10.000   
##   
## politicalChange democ autoc   
## autocratization: 70 Min. :-88.000 Min. :-88.0000   
## democratization: 155 1st Qu.: 5.000 1st Qu.: 0.0000   
## no change :4621 Median : 9.000 Median : 0.0000   
## Mean : 5.445 Mean : -0.4111   
## 3rd Qu.: 10.000 3rd Qu.: 1.0000   
## Max. : 10.000 Max. : 10.0000   
##   
## durable xrreg xrcomp xropen   
## Min. : 0.00 Min. :-88.00 Min. :-88.0000 Min. :-88.000   
## 1st Qu.: 10.00 1st Qu.: 2.00 1st Qu.: 2.0000 1st Qu.: 4.000   
## Median : 19.00 Median : 3.00 Median : 3.0000 Median : 4.000   
## Mean : 29.46 Mean : 1.08 Mean : 0.7994 Mean : 1.973   
## 3rd Qu.: 48.00 3rd Qu.: 3.00 3rd Qu.: 3.0000 3rd Qu.: 4.000   
## Max. :208.00 Max. : 3.00 Max. : 3.0000 Max. : 4.000   
##   
## xconst parreg parcomp exrec   
## Min. :-88.000 Min. :-88.000 Min. :-88.000 Min. :-88.000   
## 1st Qu.: 5.000 1st Qu.: 2.000 1st Qu.: 3.000 1st Qu.: 7.000   
## Median : 7.000 Median : 4.000 Median : 4.000 Median : 8.000   
## Mean : 4.001 Mean : 2.175 Mean : 2.311 Mean : 5.236   
## 3rd Qu.: 7.000 3rd Qu.: 5.000 3rd Qu.: 5.000 3rd Qu.: 8.000   
## Max. : 7.000 Max. : 5.000 Max. : 5.000 Max. : 8.000   
##   
## exconst polcomp pr cl   
## Min. :-88.000 Min. :-88.000 Min. :1.000 Min. :1.000   
## 1st Qu.: 5.000 1st Qu.: 7.000 1st Qu.:1.000 1st Qu.:1.000   
## Median : 7.000 Median : 9.000 Median :2.000 Median :2.000   
## Mean : 4.002 Mean : 6.316 Mean :2.726 Mean :2.733   
## 3rd Qu.: 7.000 3rd Qu.: 10.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. : 7.000 Max. : 10.000 Max. :7.000 Max. :7.000   
##   
## sum mean status inverse\_pr   
## Min. : 2.000 Min. :1.000 Free :2913 Min. :1.000   
## 1st Qu.: 2.000 1st Qu.:1.000 Not Free : 855 1st Qu.:4.000   
## Median : 4.000 Median :2.000 Partly Free:1078 Median :6.000   
## Mean : 5.507 Mean :2.748 Mean :5.275   
## 3rd Qu.: 8.000 3rd Qu.:4.000 3rd Qu.:7.000   
## Max. :14.000 Max. :7.000 Max. :7.000   
##   
## inverse\_cl inverse\_mean politicalChangeFH region   
## Min. :1.000 Min. :1.000 autocratization: 242 Africa :1031   
## 1st Qu.:4.000 1st Qu.:4.000 democratization: 296 Americas: 950   
## Median :6.000 Median :6.000 no change :4308 Asia : 981   
## Mean :5.269 Mean :5.248 Europe :1320   
## 3rd Qu.:7.000 3rd Qu.:7.000 Oceania : 564   
## Max. :7.000 Max. :7.000   
##   
## subregion   
## Caribbean : 467   
## Western Europe : 371   
## Eastern Africa : 370   
## Western Asia : 351   
## Micronesia : 339   
## Southern Europe: 322   
## (Other) :2626

From the summary statistics of the other variables, we can see that, among the Polity IV variables, there are some extreme values of -66 (for interruption periods), -77 (for interregnum), and -88 (for transition periods), which are known as “standardized authority codes” (See Marshall, M. G., Gurr, T. R., & Jaggers, K., 2018). Since there is no commonly agreed procedure on how to handle these extreme values among scholars (See Plümper, T., & Neumayer, E., 2010), we decided to remove them from our data, because they may falsify our results.

# Removing the interruption periods (-66), interregnum periods (-77), and transitional periods (-88)  
  
WGIdevRegimeType <- filter(WGIdevRegimeType, democ != -88 & democ != -77 & democ != -66)  
  
summary(WGIdevRegimeType)

## X country M49Code iso2c   
## Min. : 1 Sudan : 23 Min. : 4.0 FI : 352   
## 1st Qu.:1208 Albania : 22 1st Qu.:218.0 LI : 138   
## Median :2433 Algeria : 22 Median :414.0 SS : 123   
## Mean :2428 Andorra : 22 Mean :415.6 TW : 73   
## 3rd Qu.:3646 Antigua and Barbuda: 22 3rd Qu.:624.0 NZ : 67   
## Max. :4846 Argentina : 22 Max. :894.0 CK : 48   
## (Other) :4614 (Other):3946   
## iso3c date stability stabilityDummy  
## FIN : 362 Min. :1996 Min. :-2.9741 0:1682   
## LIE : 143 1st Qu.:2001 1st Qu.:-0.3903 1:3065   
## SSD : 116 Median :2007 Median : 0.4939   
## NZL : 69 Mean :2007 Mean : 0.2708   
## NLD : 42 3rd Qu.:2012 3rd Qu.: 1.0898   
## GRL : 36 Max. :2017 Max. : 1.9651   
## (Other):3979   
## stabilityCategory corruptionControl governmentEffectiveness  
## Highly Stable :1498 Min. :-1.8257 Min. :-2.4459   
## Highly Unstable : 592 1st Qu.:-0.5993 1st Qu.:-0.5802   
## Moderately Stable :1567 Median : 0.2181 Median : 0.2478   
## Moderately Unstable:1090 Mean : 0.3243 Mean : 0.3686   
## 3rd Qu.: 1.3499 3rd Qu.: 1.5083   
## Max. : 2.4700 Max. : 2.4370   
##   
## regulatoryQuality ruleOfLaw voiceAndAccountability  
## Min. :-2.5296 Min. :-2.4234 Min. :-2.3134   
## 1st Qu.:-0.5105 1st Qu.:-0.6180 1st Qu.:-0.5104   
## Median : 0.3194 Median : 0.3376 Median : 0.4678   
## Mean : 0.3235 Mean : 0.2555 Mean : 0.2479   
## 3rd Qu.: 1.3565 3rd Qu.: 1.1024 3rd Qu.: 1.1133   
## Max. : 2.2605 Max. : 2.1003 Max. : 1.8010   
##   
## GNIperCapita devCategory GDPannualGrowthRate  
## Min. : 204.9 High Income :1452 Min. :-52.428   
## 1st Qu.: 1351.3 Low Income : 941 1st Qu.: 1.665   
## Median : 4162.2 Lower Middle Income:1415 Median : 3.774   
## Mean : 11831.4 Upper Middle Income: 939 Mean : 3.878   
## 3rd Qu.: 17810.4 3rd Qu.: 5.963   
## Max. :119043.8 Max. :149.973   
##   
## HDI GINI povertyHeadCount polityScore   
## Min. :0.2350 Min. :16.20 Min. : 0.00 Min. :-10.000   
## 1st Qu.:0.5780 1st Qu.:37.30 1st Qu.: 1.20 1st Qu.: 5.000   
## Median :0.6990 Median :42.10 Median : 5.70 Median : 9.000   
## Mean :0.6782 Mean :41.19 Mean :14.06 Mean : 5.978   
## 3rd Qu.:0.7950 3rd Qu.:45.05 3rd Qu.:19.90 3rd Qu.: 10.000   
## Max. :0.9530 Max. :65.80 Max. :86.00 Max. : 10.000   
##   
## polityCategory politicalChange democ autoc   
## Anocracy :1060 autocratization: 62 Min. : 0.000 Min. : 0.000   
## Autocracy: 468 democratization: 133 1st Qu.: 5.000 1st Qu.: 0.000   
## Democracy:3219 no change :4552 Median : 9.000 Median : 0.000   
## Mean : 7.158 Mean : 1.179   
## 3rd Qu.:10.000 3rd Qu.: 1.000   
## Max. :10.000 Max. :10.000   
##   
## durable xrreg xrcomp xropen   
## Min. : 0.00 Min. :1.000 Min. :0.000 Min. :0.000   
## 1st Qu.: 10.00 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:4.000   
## Median : 20.00 Median :3.000 Median :3.000 Median :4.000   
## Mean : 30.07 Mean :2.702 Mean :2.415 Mean :3.613   
## 3rd Qu.: 49.00 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:4.000   
## Max. :208.00 Max. :3.000 Max. :3.000 Max. :4.000   
##   
## xconst parreg parcomp exrec   
## Min. :1.000 Min. :1.00 Min. :0.000 Min. :1.000   
## 1st Qu.:5.000 1st Qu.:3.00 1st Qu.:3.000 1st Qu.:7.000   
## Median :7.000 Median :4.00 Median :4.000 Median :8.000   
## Mean :5.684 Mean :3.82 Mean :3.958 Mean :6.944   
## 3rd Qu.:7.000 3rd Qu.:5.00 3rd Qu.:5.000 3rd Qu.:8.000   
## Max. :7.000 Max. :5.00 Max. :5.000 Max. :8.000   
##   
## exconst polcomp pr cl   
## Min. :1.000 Min. : 1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:5.000 1st Qu.: 7.000 1st Qu.:1.000 1st Qu.:1.000   
## Median :7.000 Median : 9.000 Median :2.000 Median :2.000   
## Mean :5.684 Mean : 8.046 Mean :2.664 Mean :2.673   
## 3rd Qu.:7.000 3rd Qu.:10.000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :7.000 Max. :10.000 Max. :7.000 Max. :7.000   
##   
## sum mean status inverse\_pr   
## Min. : 2.000 Min. :1.000 Free :2913 Min. :1.000   
## 1st Qu.: 2.000 1st Qu.:1.000 Not Free : 788 1st Qu.:4.000   
## Median : 4.000 Median :2.000 Partly Free:1046 Median :6.000   
## Mean : 5.387 Mean :2.688 Mean :5.337   
## 3rd Qu.: 8.000 3rd Qu.:4.000 3rd Qu.:7.000   
## Max. :14.000 Max. :7.000 Max. :7.000   
##   
## inverse\_cl inverse\_mean politicalChangeFH region   
## Min. :1.000 Min. :1.000 autocratization: 224 Africa : 977   
## 1st Qu.:4.000 1st Qu.:4.000 democratization: 274 Americas: 940   
## Median :6.000 Median :6.000 no change :4249 Asia : 950   
## Mean :5.328 Mean :5.308 Europe :1320   
## 3rd Qu.:7.000 3rd Qu.:7.000 Oceania : 560   
## Max. :7.000 Max. :7.000   
##   
## subregion   
## Caribbean : 458   
## Western Europe : 371   
## Eastern Africa : 349   
## Micronesia : 339   
## Western Asia : 334   
## Southern Europe: 322   
## (Other) :2574

dim(WGIdevRegimeType)

## [1] 4747 46

### Step 3: Preparing the data: data randomization and feature selection

The data were arranged in a country-year format. Therefore, it is a good idea to randoming the rows before partitioning the dataset

# Randomizing the dataset  
set.seed(112)  
rows <- sample(nrow(WGIdevRegimeType))  
WGIdevRegimeTypeRand <- WGIdevRegimeType[rows, ]

# Selecting the variables to be included in the classification model. We need to remove the political stability score (stability) and stability category (stabilityCategory) from this dataset.  
  
WGIdevRegimeTypeRandClass <- select(WGIdevRegimeTypeRand, stabilityDummy, corruptionControl, governmentEffectiveness, regulatoryQuality, ruleOfLaw, voiceAndAccountability, GNIperCapita, GDPannualGrowthRate, HDI, GINI, povertyHeadCount, polityScore, polityCategory, politicalChange, democ, autoc, durable, xrreg, xrcomp, xropen, xconst, parreg, parcomp, exrec, exconst, polcomp, pr, cl, sum, mean, status, inverse\_pr, inverse\_cl, inverse\_mean, politicalChangeFH, region, subregion)

# Selecting the variables to be included in the regression model. We need to remove the stability dummy (stabilityDummy) and stability category (stabilityCategory) from this dataset.  
  
WGIdevRegimeTypeRandReg <- select(WGIdevRegimeTypeRand, stability, corruptionControl, governmentEffectiveness, regulatoryQuality, ruleOfLaw, voiceAndAccountability, GNIperCapita, GDPannualGrowthRate, HDI, GINI, povertyHeadCount, polityScore, polityCategory, politicalChange, democ, autoc, durable, xrreg, xrcomp, xropen, xconst, parreg, parcomp, exrec, exconst, polcomp, pr, cl, sum, mean, status, inverse\_pr, inverse\_cl, inverse\_mean, politicalChangeFH, region, subregion)

### Step 4: Modeling and predicting with classification

# Partitionning the dataset into training set and testing set for the classification model  
set.seed(123)  
inTrain1 <- createDataPartition(y = WGIdevRegimeTypeRandClass$stabilityDummy, p = .75, list = FALSE)  
training1 <- WGIdevRegimeTypeRandClass[inTrain1,]  
testing1 <- WGIdevRegimeTypeRandClass[-inTrain1,]

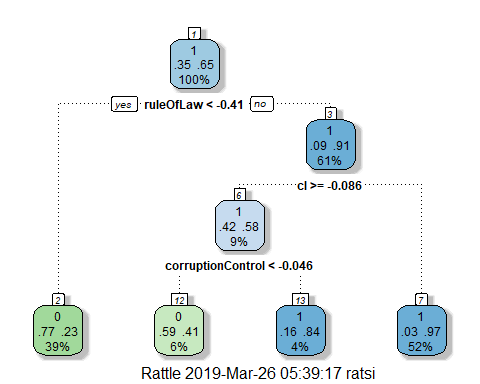
# Training the classification model  
  
classModel <- train(stabilityDummy ~ ., method = "rpart", data = training1,  
 trControl = trainControl(method = "cv", number = 5,  
 verboseIter = TRUE),  
 preProcess = c("zv", "nzv", "center", "scale")  
 )

## + Fold1: cp=0.01189   
## - Fold1: cp=0.01189   
## + Fold2: cp=0.01189   
## - Fold2: cp=0.01189   
## + Fold3: cp=0.01189   
## - Fold3: cp=0.01189   
## + Fold4: cp=0.01189   
## - Fold4: cp=0.01189   
## + Fold5: cp=0.01189   
## - Fold5: cp=0.01189   
## Aggregating results  
## Selecting tuning parameters  
## Fitting cp = 0.0119 on full training set

print(classModel$finalModel)

## n= 3561   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 3561 1262 1 (0.35439483 0.64560517)   
## 2) ruleOfLaw< -0.4129169 1391 314 0 (0.77426312 0.22573688) \*  
## 3) ruleOfLaw>=-0.4129169 2170 185 1 (0.08525346 0.91474654)   
## 6) cl>=-0.08551258 327 137 1 (0.41896024 0.58103976)   
## 12) corruptionControl< -0.04627442 196 80 0 (0.59183673 0.40816327) \*  
## 13) corruptionControl>=-0.04627442 131 21 1 (0.16030534 0.83969466) \*  
## 7) cl< -0.08551258 1843 48 1 (0.02604449 0.97395551) \*

# Plotting the classification model  
fancyRpartPlot(classModel$finalModel)



# Making prediction and assessing the model performance  
  
classPred <- predict(classModel, testing1)  
  
confusionMatrix(classPred, testing1$stabilityDummy)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 394 135  
## 1 26 631  
##   
## Accuracy : 0.8642   
## 95% CI : (0.8434, 0.8832)  
## No Information Rate : 0.6459   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7197   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9381   
## Specificity : 0.8238   
## Pos Pred Value : 0.7448   
## Neg Pred Value : 0.9604   
## Prevalence : 0.3541   
## Detection Rate : 0.3322   
## Detection Prevalence : 0.4460   
## Balanced Accuracy : 0.8809   
##   
## 'Positive' Class : 0   
##

### Step 5: Modeling and predicting with regression

# Partitionning the dataset into training set and testing set for the regression model  
set.seed(234)  
inTrain2 <- createDataPartition(y = WGIdevRegimeTypeRandReg$stability, p = .75, list = FALSE)  
training2 <- WGIdevRegimeTypeRandReg[inTrain2,]  
testing2 <- WGIdevRegimeTypeRandReg[-inTrain2,]  
dim(training2)

## [1] 3563 37

dim(testing2)

## [1] 1184 37

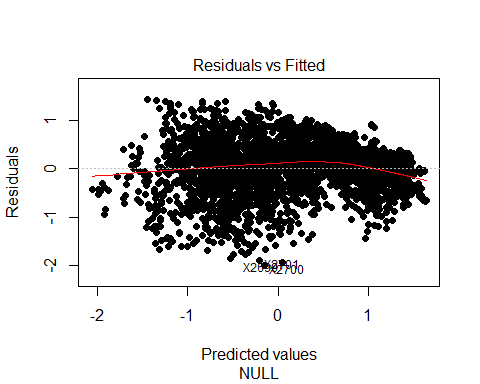
# Training the regression model  
  
regModel <- train(stability ~ ., method = "glm", data = training2,  
 trControl = trainControl(method = "cv", number = 5,  
 verboseIter = TRUE),  
 preProcess = c("zv", "nzv", "center", "scale", "pca")  
 )

## + Fold1: parameter=none   
## - Fold1: parameter=none   
## + Fold2: parameter=none   
## - Fold2: parameter=none   
## + Fold3: parameter=none   
## - Fold3: parameter=none   
## + Fold4: parameter=none   
## - Fold4: parameter=none   
## + Fold5: parameter=none   
## - Fold5: parameter=none   
## Aggregating results  
## Fitting final model on full training set

regFinalModel <- regModel$finalModel  
   
print(regModel)

## Generalized Linear Model   
##   
## 3563 samples  
## 36 predictor  
##   
## Pre-processing: centered (48), scaled (48), principal component  
## signal extraction (48), remove (15)   
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 2851, 2851, 2851, 2850, 2849   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 0.5108481 0.7221264 0.3855061

# Regression model diagnostics  
  
plot(regFinalModel, 1, pch = 19)



# Making prediction and assessing the model performance  
  
regPred <- predict(regModel, testing2)  
  
postResample(pred = regPred, obs = testing2$stability)

## RMSE Rsquared MAE   
## 0.5044487 0.7318404 0.3813083

### Step 6: Conclusion