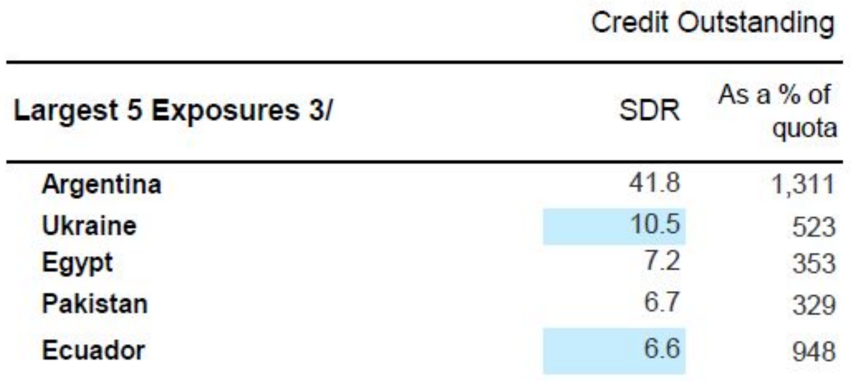
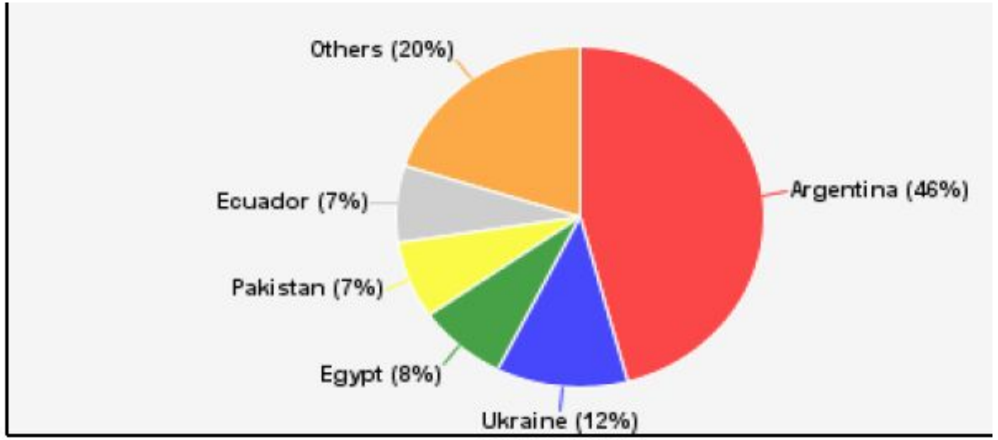
Evaluación del cumplimiento de los objetivos fiscales y externos en los programas del FMI: un enfoque basado en el aprendizaje automático

Motivación: sólo 5 países explican el 80% de la cartera de créditos del FMI…



	Credit Outstanding	
Largest 5 Exposures 3/	SDR	As a % of quota
Argentina	41.8	1,311
Ukraine	10.5	523
Egypt	7.2	353
Pakistan	6.7	329
Ecuador	6.6	948



Fuente: https://www.imf.org/en/Data/IMF-Finances

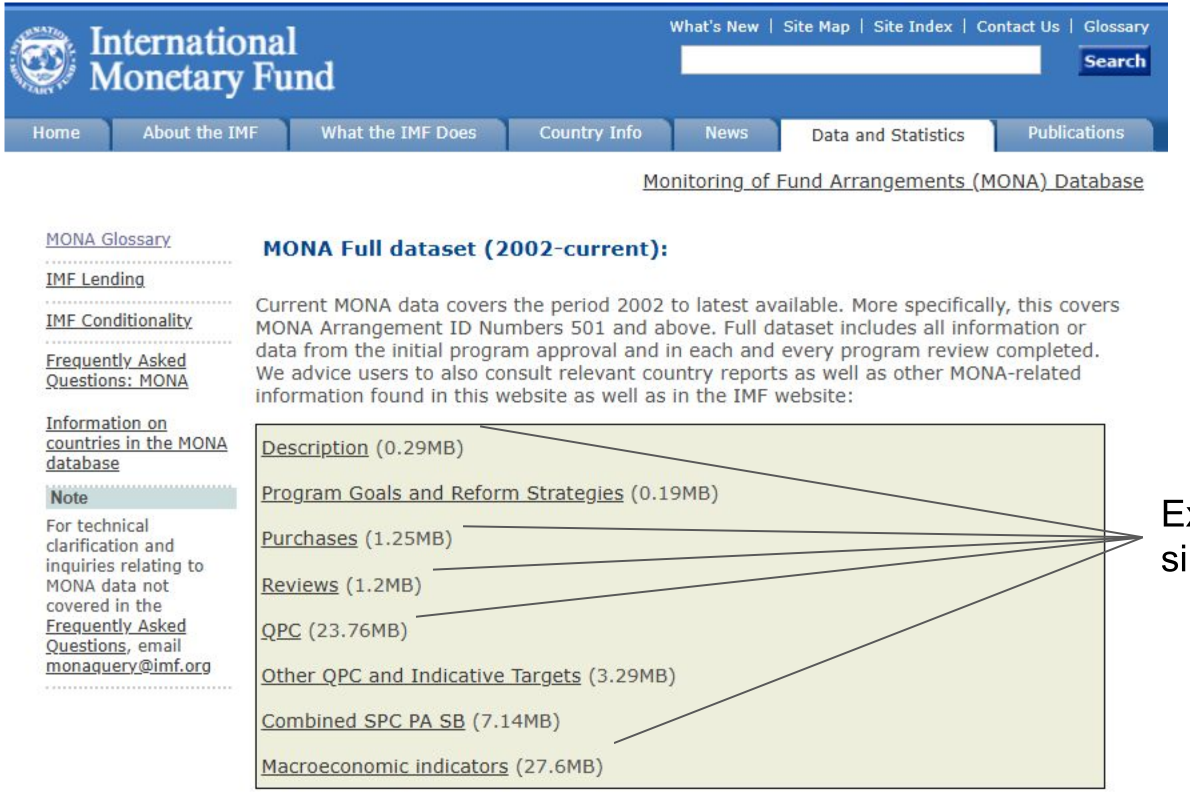
Preguntas de investigación

1. ¿**Supera** el modelo **Random Forest (RF)** al modelo benchmark (**regresión logística regularizada con Lasso**) en la predicción out-of-sample del incumplimiento de metas cuantitativas de desempeño (QPCs) fiscales y externas de los programas del FMI?

2. ¿Presenta RF una mayor capacidad predictiva para las **metas fiscales o para las metas externas**?

3. ¿Cuál es el desempeño predictivo del modelo RF para el **caso de los principales deudores del FMI**: Argentina, Ucrania, Egipto, Pakistán y Ecuador?

MONA "dataset"


International Monetary Fund
What's New | Site Map | Site Index | Contact Us | Glossary
Search
Home
About the IMF
What the IMF Does
Country Info
News
Data and Statistics
Publications
Monitoring of Fund Arrangements (MONA) Database
MONA Glossary
IMF Lending
IMF Conditionality
Frequently Asked Questions: MONA
Information on countries in the MONA database
Note

For technical clarification and inquiries relating to MONA data not covered in the Frequently Asked Questions, email monaquery@imf.org

MONA Full dataset (2002-current):

Current MONA data covers the period 2002 to latest available. More specifically, this covers MONA Arrangement ID Numbers 501 and above. Full dataset includes all information or data from the initial program approval and in each and every program review completed. We advice users to also consult relevant country reports as well as other MONA-related information found in this website as well as in the IMF website:

Description (0.29MB)

Program Goals and Reform Strategies (0.19MB)

Purchases (1.25MB)

Reviews (1.2MB)

QPC (23.76MB)

Other QPC and Indicative Targets (3.29MB)

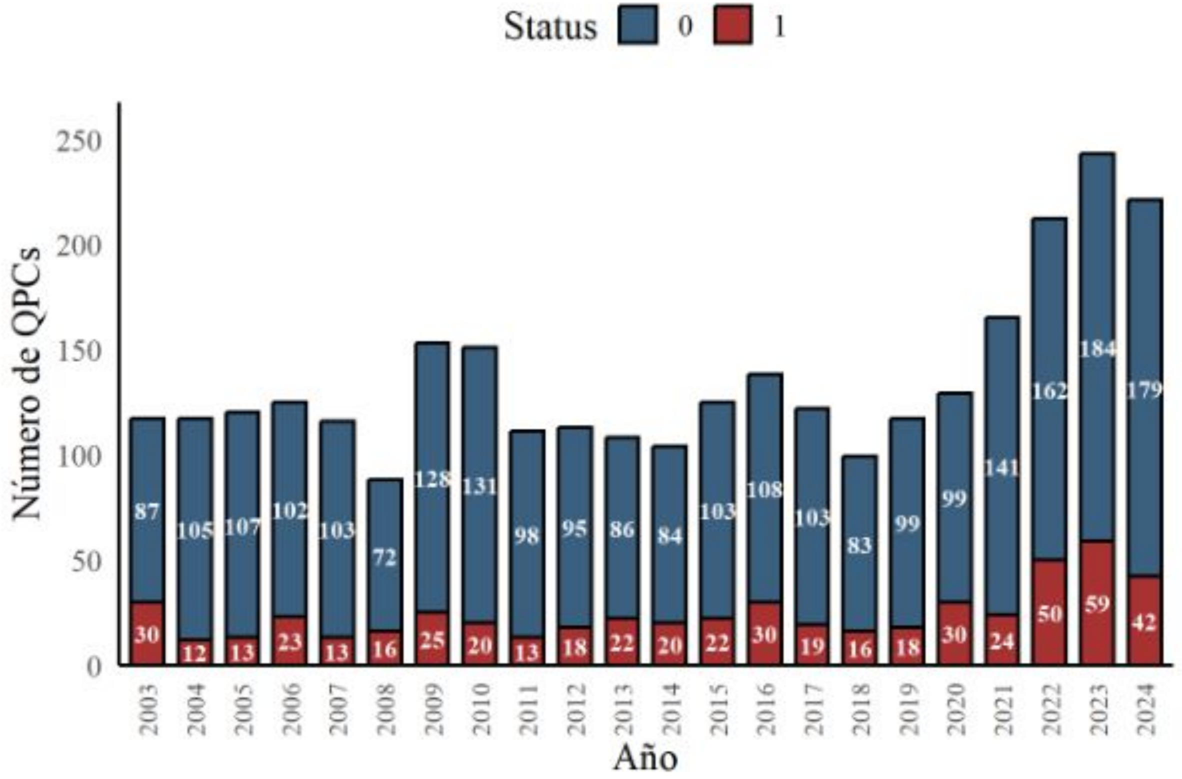
Combined SPC PA SB (7.14MB)

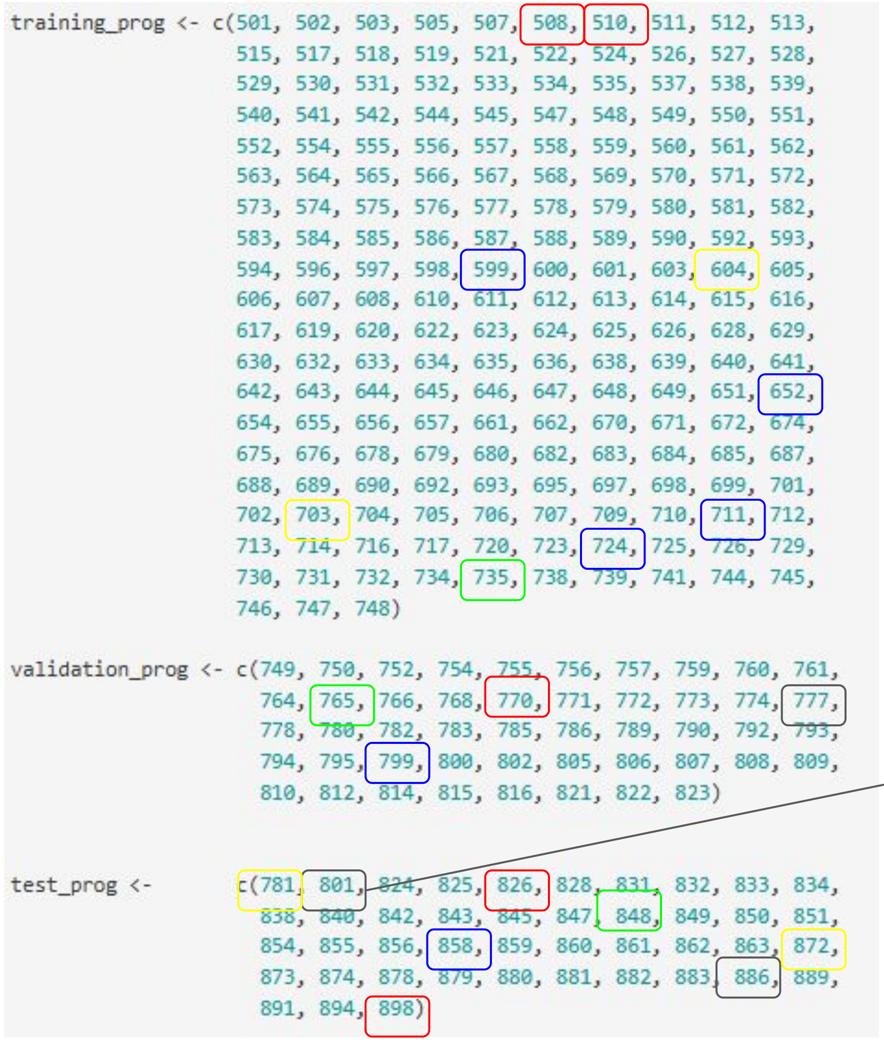
Macroeconomic indicators (27.6MB)

Excels
sin key común

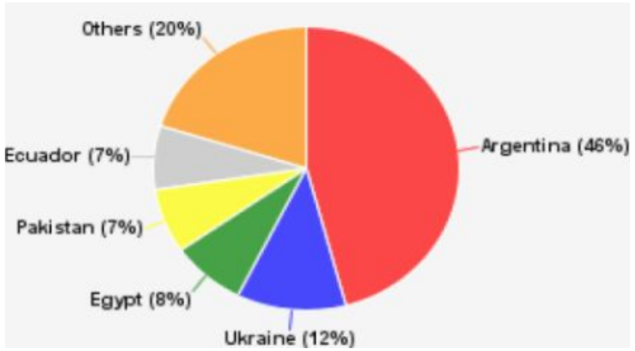
https://www.imf.org/external/np/pdr/mona/index.aspx

Labels: not-met QPCs (1) vs. met (0)





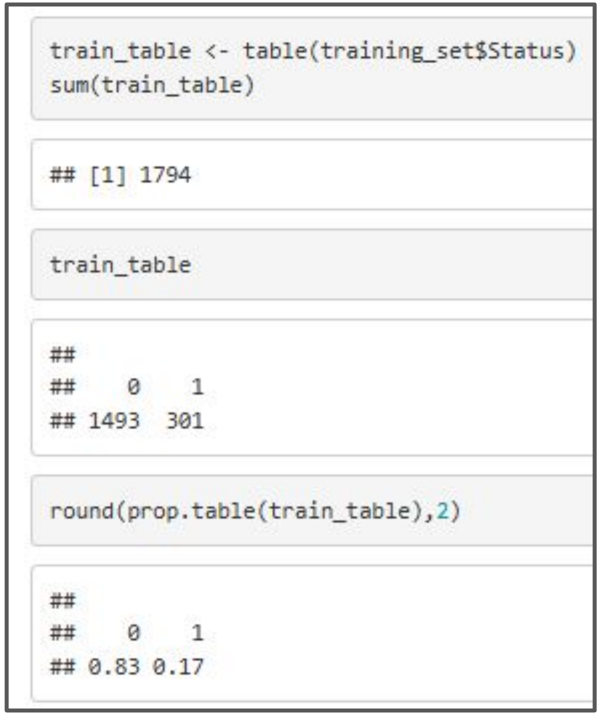
```
training_prog <- c(501, 502, 503, 505, 507, 508, 510, 511, 512, 513,
                   515, 517, 518, 519, 521, 522, 524, 526, 527, 528,
                   529, 530, 531, 532, 533, 534, 535, 537, 538, 539,
                   540, 541, 542, 544, 545, 547, 548, 549, 550, 551,
                   552, 554, 555, 556, 557, 558, 559, 560, 561, 562,
                   563, 564, 565, 566, 567, 568, 569, 570, 571, 572,
                   573, 574, 575, 576, 577, 578, 579, 580, 581, 582,
                   583, 584, 585, 586, 587, 588, 589, 590, 592, 593,
                   594, 596, 597, 598, 599, 600, 601, 603, 604, 605,
                   606, 607, 608, 610, 611, 612, 613, 614, 615, 616,
                   617, 619, 620, 622, 623, 624, 625, 626, 628, 629,
                   630, 632, 633, 634, 635, 636, 638, 639, 640, 641,
                   642, 643, 644, 645, 646, 647, 648, 649, 651, 652,
                   654, 655, 656, 657, 661, 662, 670, 671, 672, 674,
                   675, 676, 678, 679, 680, 682, 683, 684, 685, 687,
                   688, 689, 690, 692, 693, 695, 697, 698, 699, 701,
                   702, 703, 704, 705, 706, 707, 709, 710, 711, 712,
                   713, 714, 716, 717, 720, 723, 724, 725, 726, 729,
                   730, 731, 732, 734, 735, 738, 739, 741, 744, 745,
                   746, 747, 748)

validation_prog <- c(749, 750, 752, 754, 755, 756, 757, 759, 760, 761,
                     764, 765, 766, 768, 770, 771, 772, 773, 774, 777,
                     778, 780, 782, 783, 785, 786, 789, 790, 792, 793,
                     794, 795, 799, 800, 802, 805, 806, 807, 808, 809,
                     810, 812, 814, 815, 816, 821, 822, 823)


test_prog <-       c(781, 801, 824, 825, 826, 828, 831, 832, 833, 834,
                     838, 840, 842, 843, 845, 847, 848, 849, 850, 851,
                     854, 855, 856, 858, 859, 860, 861, 862, 863, 872,
                     873, 874, 878, 879, 880, 881, 882, 883, 886, 889,
                     891, 894, 898)
```



Se mantiene orden de programas.

2 excepciones: se pasa de validation a train los programas 801 (Ecuador) y 781 (Pakistan) dada las pocas observaciones en test de ambos países.

Train - validation - test sets



```
train_table <- table(training_set$Status)
sum(train_table)
```

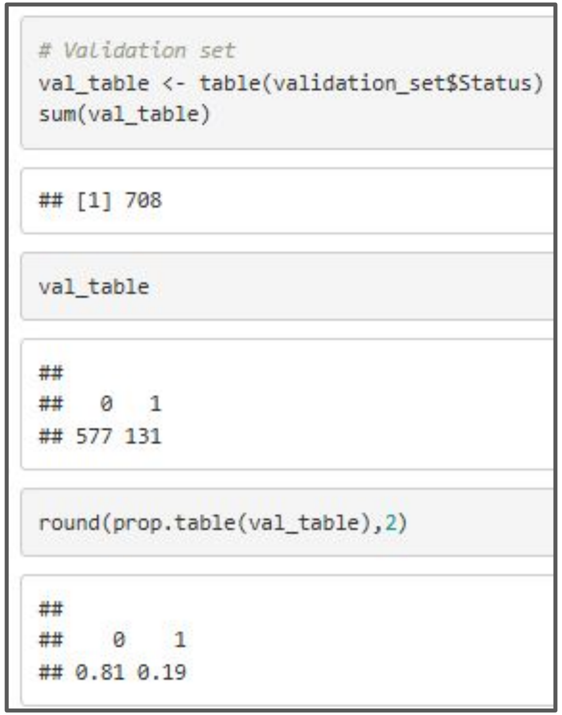
```
## [1] 1794
```

```
train_table
```

```
##
##    0    1
## 1493  301
```

```
round(prop.table(train_table),2)
```

```
##
##    0    1
## 0.83 0.17
```



```
# Validation set
val_table <- table(validation_set$Status)
sum(val_table)
```

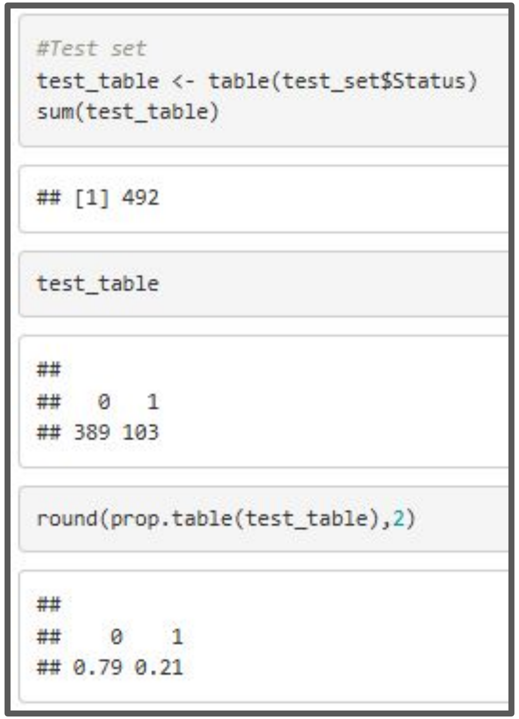
```
## [1] 708
```

```
val_table
```

```
##
##   0   1
## 577 131
```

```
round(prop.table(val_table),2)
```

```
##
##    0    1
## 0.81 0.19
```



```
#Test set
test_table <- table(test_set$Status)
sum(test_table)
```

```
## [1] 492
```

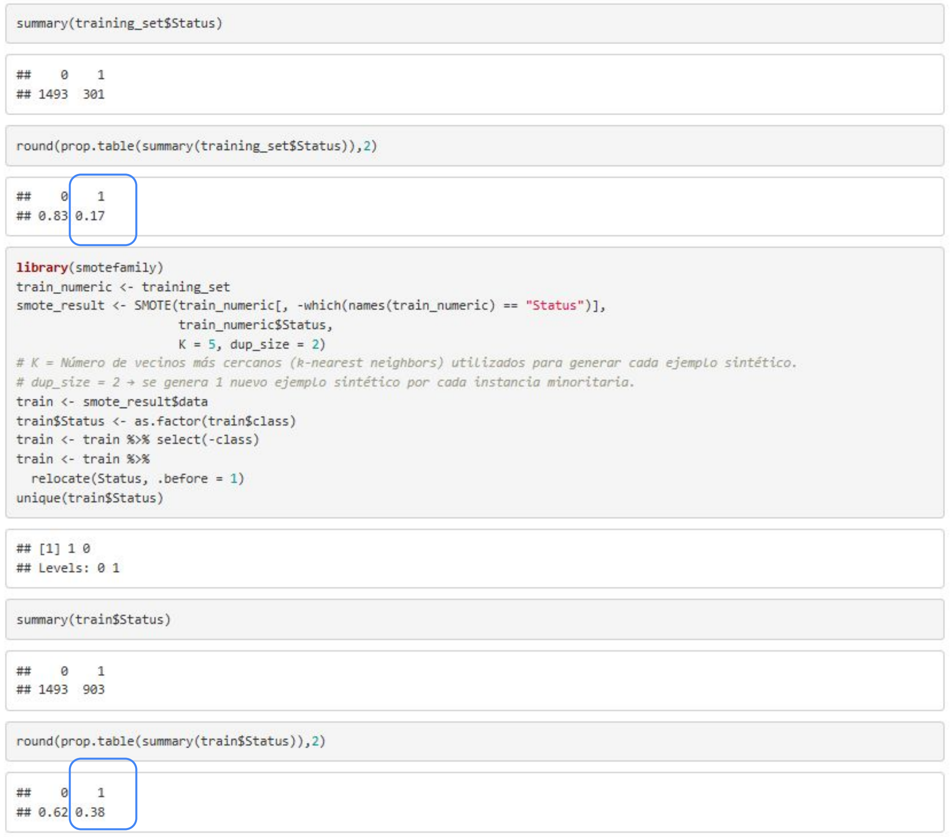
```
test_table
```

```
##
##   0   1
## 389 103
```

```
round(prop.table(test_table),2)
```

```
##
##    0    1
## 0.79 0.21
```

SMOTE para equilibrar el train set



```
summary(training_set$Status)
```

```
##    0    1
## 1493  301
```

```
round(prop.table(summary(training_set$Status)),2)
```

```
##    0    1
## 0.83 0.17
```

```
library(smotefamily)
train_numeric <- training_set
smote_result <- SMOTE(train_numeric[, -which(names(train_numeric) == "Status")],
                      train_numeric$Status,
                      K = 5, dup_size = 2)
# K = Número de vecinos más cercanos (k-nearest neighbors) utilizados para generar cada ejemplo sintético.
# dup_size = 2 → se genera 1 nuevo ejemplo sintético por cada instancia minoritaria.
train <- smote_result$data
train$Status <- as.factor(train$class)
train <- train %>% select(-class)
train <- train %>%
  relocate(Status, .before = 1)
unique(train$Status)
```

```
## [1] 1 0
## Levels: 0 1
```

```
summary(train$Status)
```

```
##    0    1
## 1493  903
```

```
round(prop.table(summary(train$Status)),2)
```

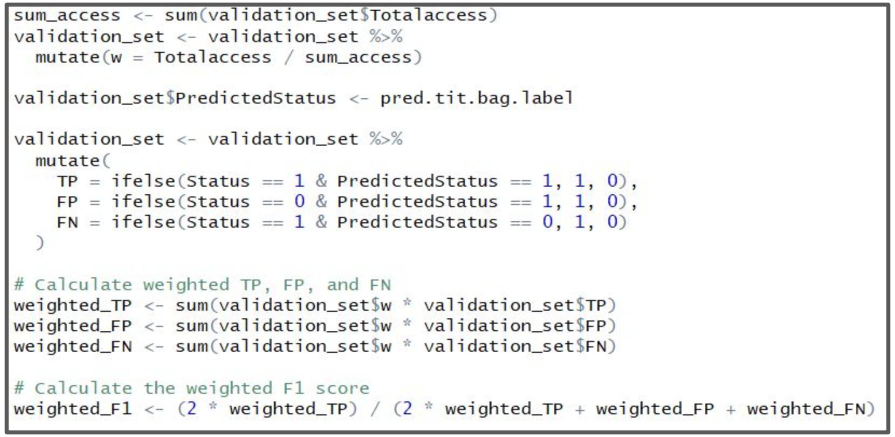
```
##    0    1
## 0.62 0.38
```

Métricas de evaluación: tradicionales + F1-Weighted

$$\text{F1 score weighted} = \frac{2 \cdot \sum_{i=1}^{i} w_i \cdot \text{TP}_i}{2 \cdot \sum_{j=1}^{j} w_j \cdot \text{TP}_j + \sum_{l=1}^{l} w_l \cdot \text{FP}_l + \sum_{t=1}^{t} w_t \cdot \text{FN}_t}$$

where

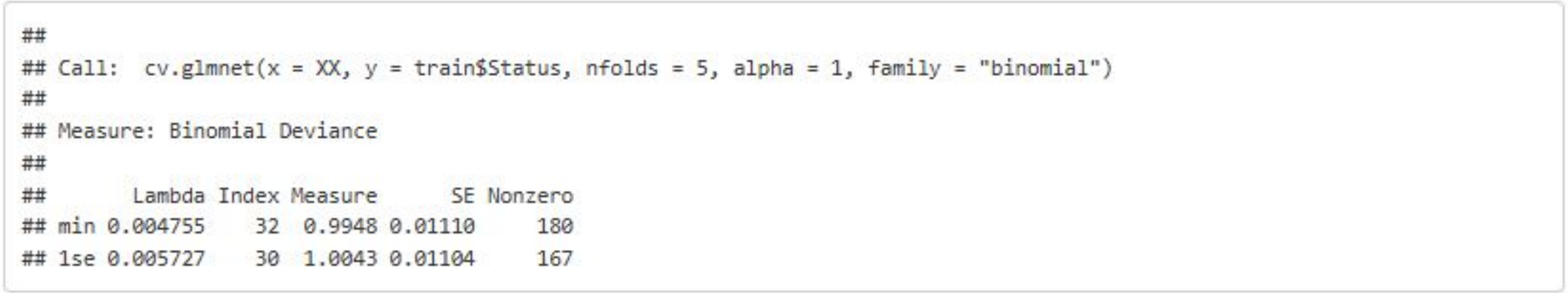
$$w_i = \frac{C_i}{\sum_{k=1}^{K} C_k}$$



```r
sum_access <- sum(validation_set$Totalaccess)
validation_set <- validation_set %>%
  mutate(w = Totalaccess / sum_access)

validation_set$PredictedStatus <- pred.tit.bag.label

validation_set <- validation_set %>%
  mutate(
    TP = ifelse(Status == 1 & PredictedStatus == 1, 1, 0),
    FP = ifelse(Status == 0 & PredictedStatus == 1, 1, 0),
    FN = ifelse(Status == 1 & PredictedStatus == 0, 1, 0)
  )

# Calculate weighted TP, FP, and FN
weighted_TP <- sum(validation_set$w * validation_set$TP)
weighted_FP <- sum(validation_set$w * validation_set$FP)
weighted_FN <- sum(validation_set$w * validation_set$FN)

# Calculate the weighted F1 score
weighted_F1 <- (2 * weighted_TP) / (2 * weighted_TP + weighted_FP + weighted_FN)
```

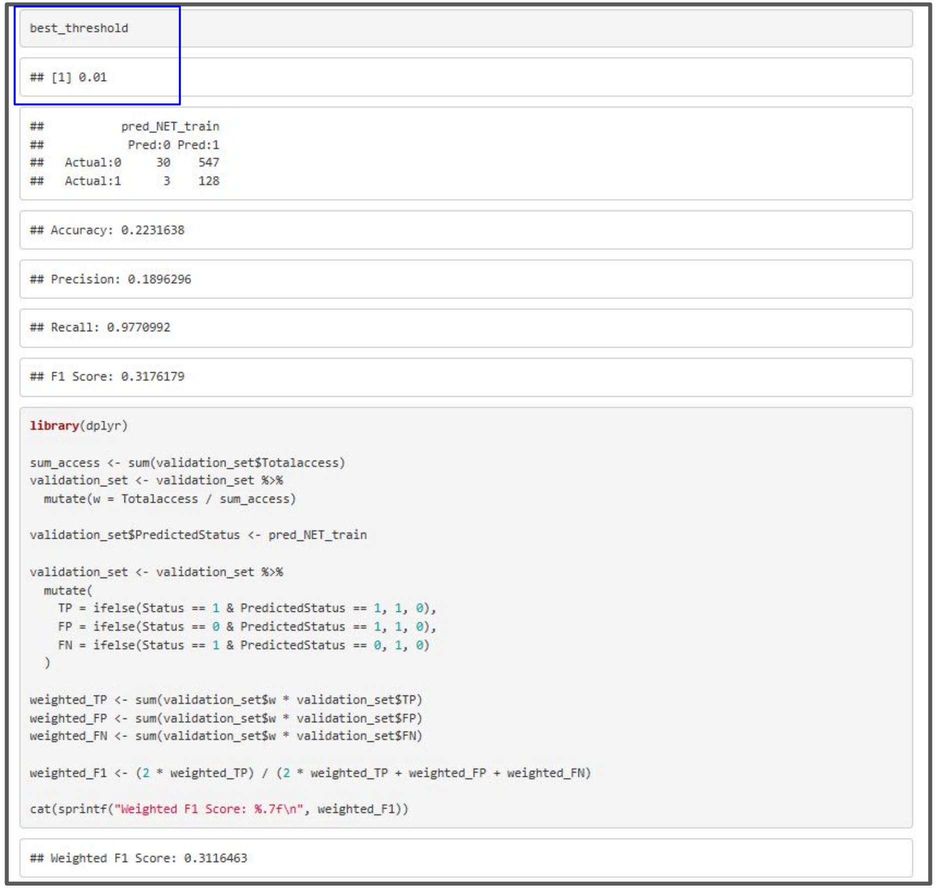
w: importancia relativa del programa (y de sus QPCs) con el país i en términos de monto pre-aprobado por el FMI.

Modelo Logit-Lasso (modelo benchmark)



```
##
## Call:  cv.glmnet(x = XX, y = train$Status, nfolds = 5, alpha = 1, family = "binomial")
##
## Measure: Binomial Deviance
##
##       Lambda Index Measure      SE Nonzero
## min 0.004755    32  0.9948 0.01110     180
## 1se 0.005727    30  1.0043 0.01104     167
```

Obtención del threshold óptimo sobre validation set



```
best_threshold
```

```
## [1] 0.01
```

```
##          pred_NET_train
##           Pred:0 Pred:1
##   Actual:0     30    547
##   Actual:1      3    128
```

```
## Accuracy: 0.2231638
```

```
## Precision: 0.1896296
```

```
## Recall: 0.9770992
```

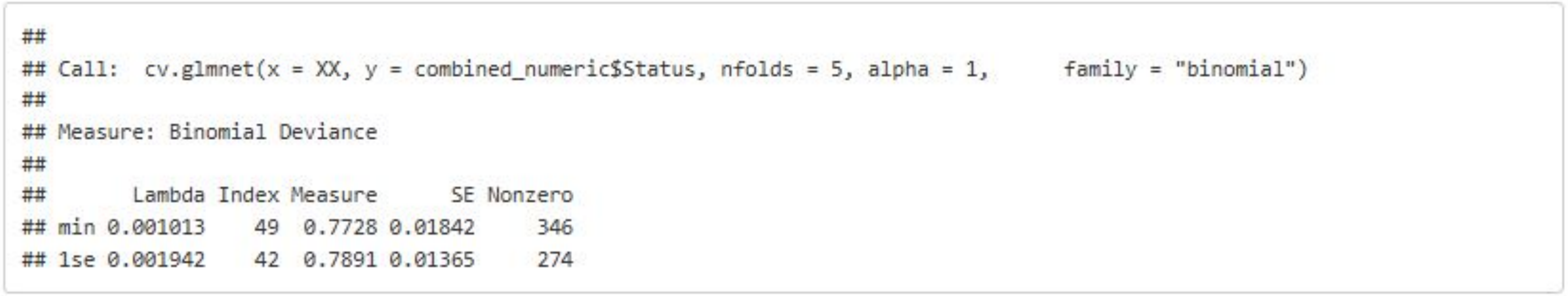
```
## F1 Score: 0.3176179
```

```r
library(dplyr)

sum_access <- sum(validation_set$Totalaccess)
validation_set <- validation_set %>%
  mutate(w = Totalaccess / sum_access)

validation_set$PredictedStatus <- pred_NET_train

validation_set <- validation_set %>%
  mutate(
    TP = ifelse(Status == 1 & PredictedStatus == 1, 1, 0),
    FP = ifelse(Status == 0 & PredictedStatus == 1, 1, 0),
    FN = ifelse(Status == 1 & PredictedStatus == 0, 1, 0)
  )

weighted_TP <- sum(validation_set$w * validation_set$TP)
weighted_FP <- sum(validation_set$w * validation_set$FP)
weighted_FN <- sum(validation_set$w * validation_set$FN)

weighted_F1 <- (2 * weighted_TP) / (2 * weighted_TP + weighted_FP + weighted_FN)

cat(sprintf("Weighted F1 Score: %.7f\n", weighted_F1))
```

```
## Weighted F1 Score: 0.3116463
```

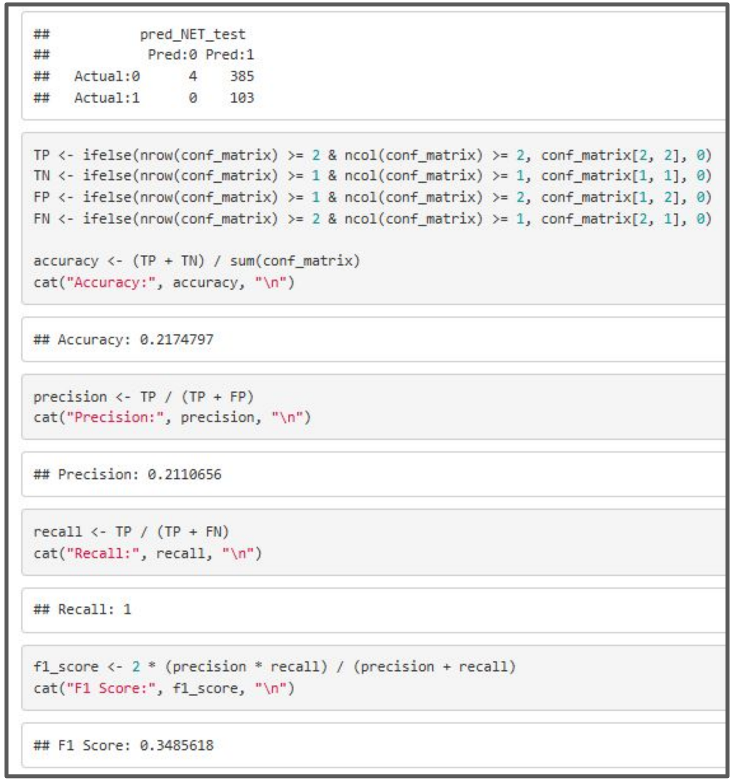
Modelo Logit-Lasso

(re-entrenado sobre el nuevo train set)



```
##
## Call:  cv.glmnet(x = XX, y = combined_numeric$Status, nfolds = 5, alpha = 1,      family = "binomial")
##
## Measure: Binomial Deviance
##
##       Lambda Index Measure      SE Nonzero
## min 0.001013    49  0.7728 0.01842     346
## 1se 0.001942    42  0.7891 0.01365     274
```

Performance en test set (out of sample)



```
##           pred_NET_test
##            Pred:0 Pred:1
##   Actual:0      4    385
##   Actual:1      0    103
```

```
TP <- ifelse(nrow(conf_matrix) >= 2 & ncol(conf_matrix) >= 2, conf_matrix[2, 2], 0)
TN <- ifelse(nrow(conf_matrix) >= 1 & ncol(conf_matrix) >= 1, conf_matrix[1, 1], 0)
FP <- ifelse(nrow(conf_matrix) >= 1 & ncol(conf_matrix) >= 2, conf_matrix[1, 2], 0)
FN <- ifelse(nrow(conf_matrix) >= 2 & ncol(conf_matrix) >= 1, conf_matrix[2, 1], 0)

accuracy <- (TP + TN) / sum(conf_matrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.2174797
```

```
precision <- TP / (TP + FP)
cat("Precision:", precision, "\n")
```

```
## Precision: 0.2110656
```

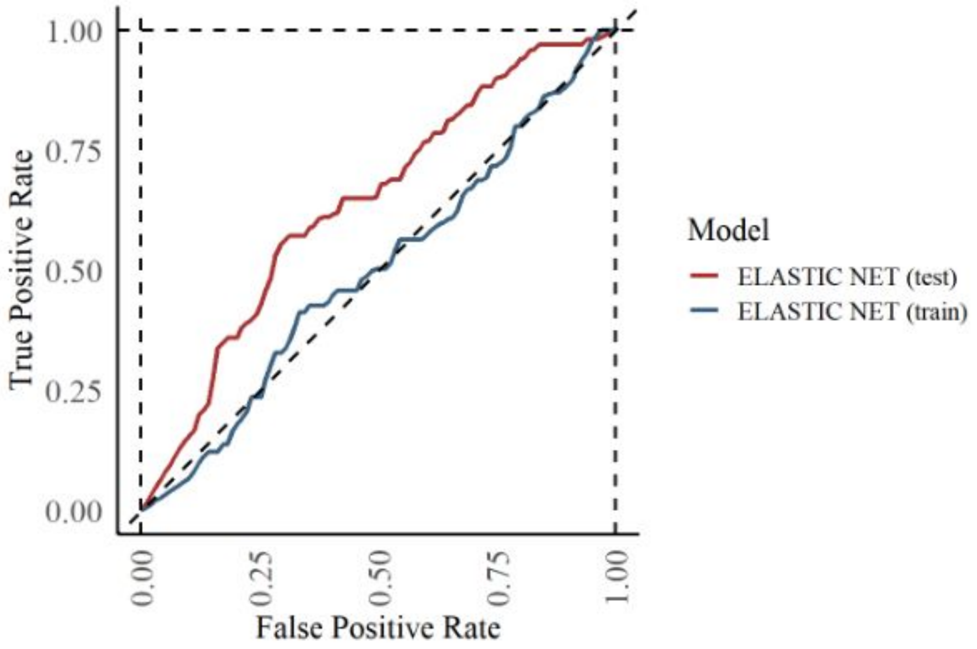
```
recall <- TP / (TP + FN)
cat("Recall:", recall, "\n")
```

```
## Recall: 1
```

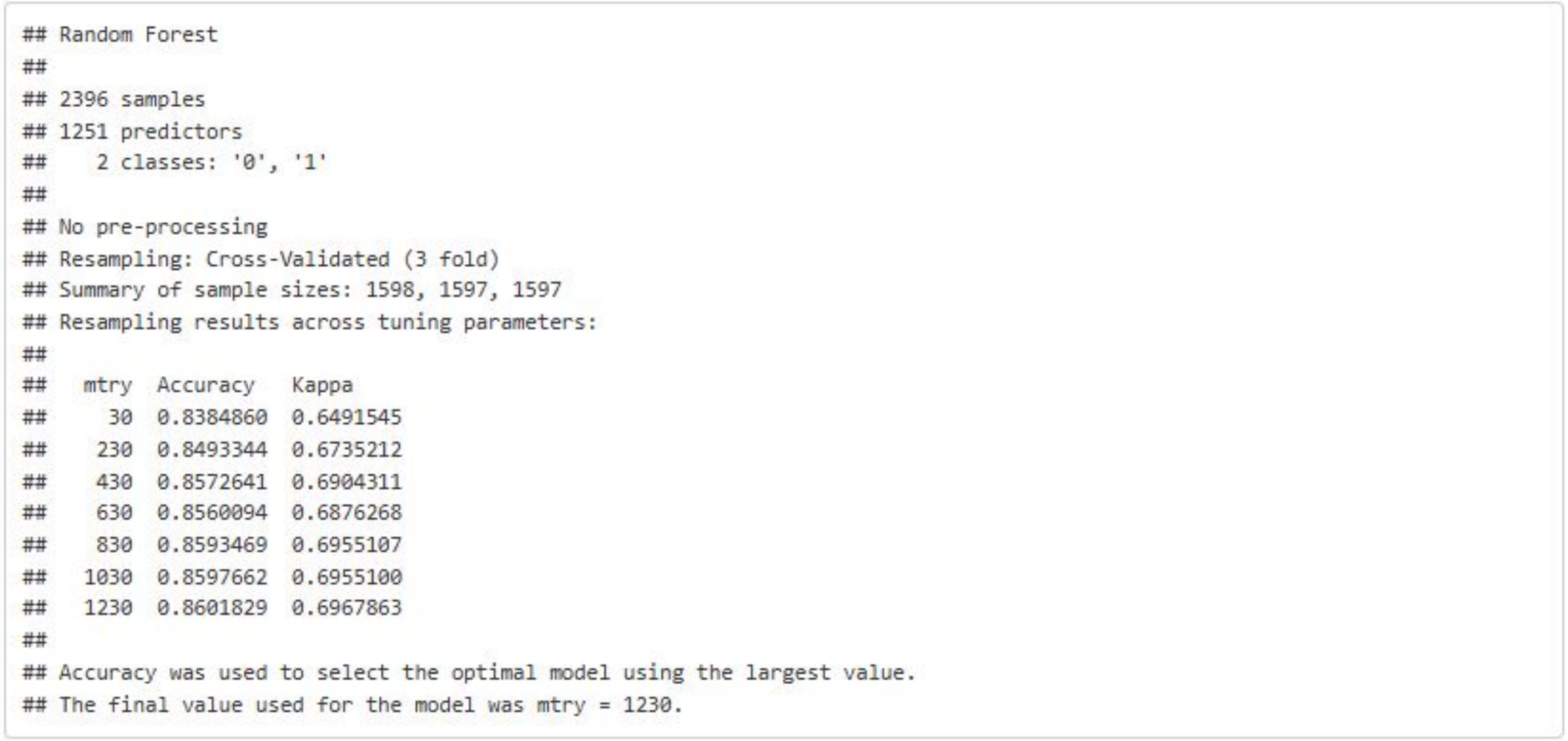
```
f1_score <- 2 * (precision * recall) / (precision + recall)
cat("F1 Score:", f1_score, "\n")
```

```
## F1 Score: 0.3485618
```

```
## Weighted F1 Score: 0.5963805
```



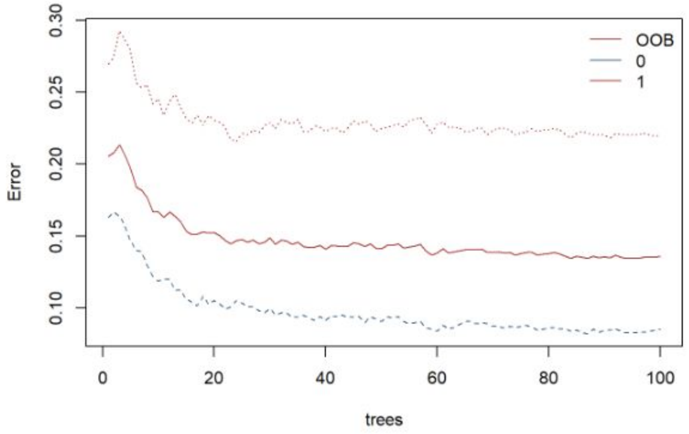
Modelo Random Forest



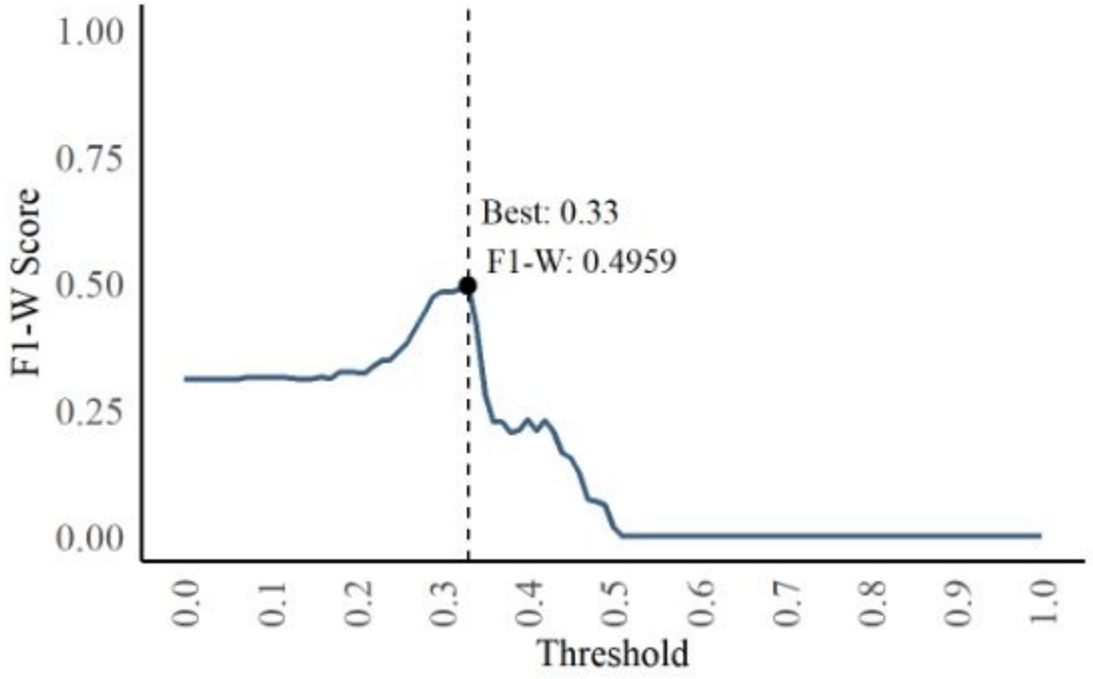
```
## Random Forest
##
## 2396 samples
## 1251 predictors
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 1598, 1597, 1597
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##     30  0.8384860  0.6491545
##    230  0.8493344  0.6735212
##    430  0.8572641  0.6904311
##    630  0.8560094  0.6876268
##    830  0.8593469  0.6955107
##   1030  0.8597662  0.6955100
##   1230  0.8601829  0.6967863
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 1230.
```

RF ajustado en train

```
## 
## Call:
##  randomForest(formula = Status ~ ., data = train, ntree = 100,      mtry = mtry_cv, importance = TRUE) 
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 1230
## 
##         OOB estimate of  error rate: 13.61%
## Confusion matrix:
##      0   1 class.error
## 0 1366 127  0.08506363
## 1  199 704  0.22037652
```

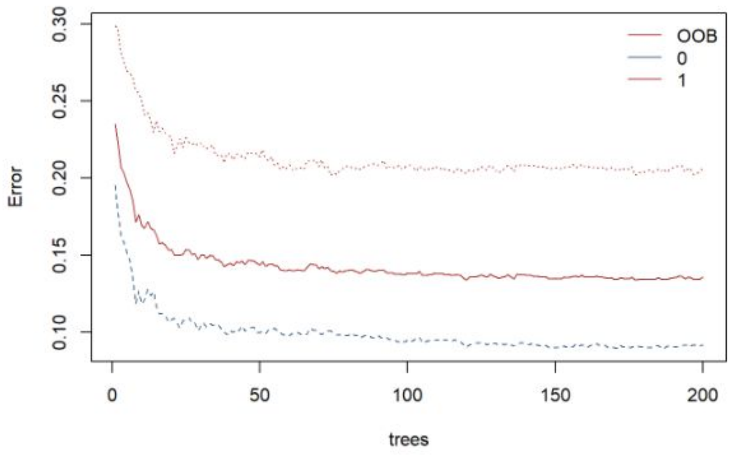


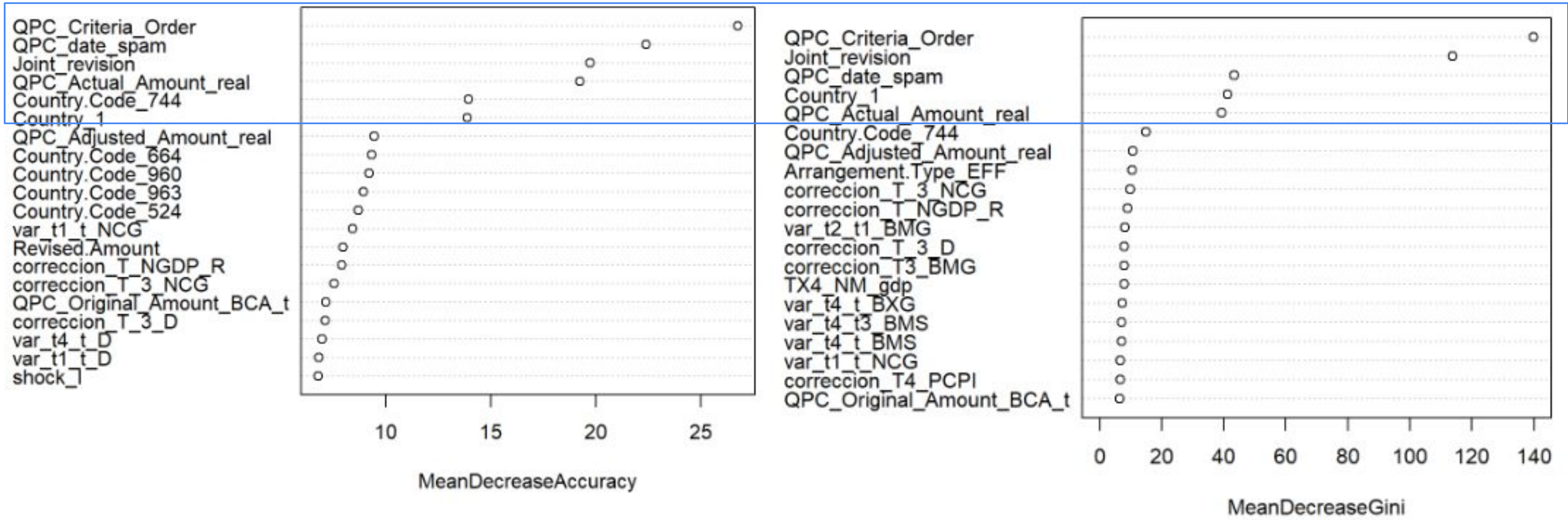
Obtención del threshold óptimo sobre validation set



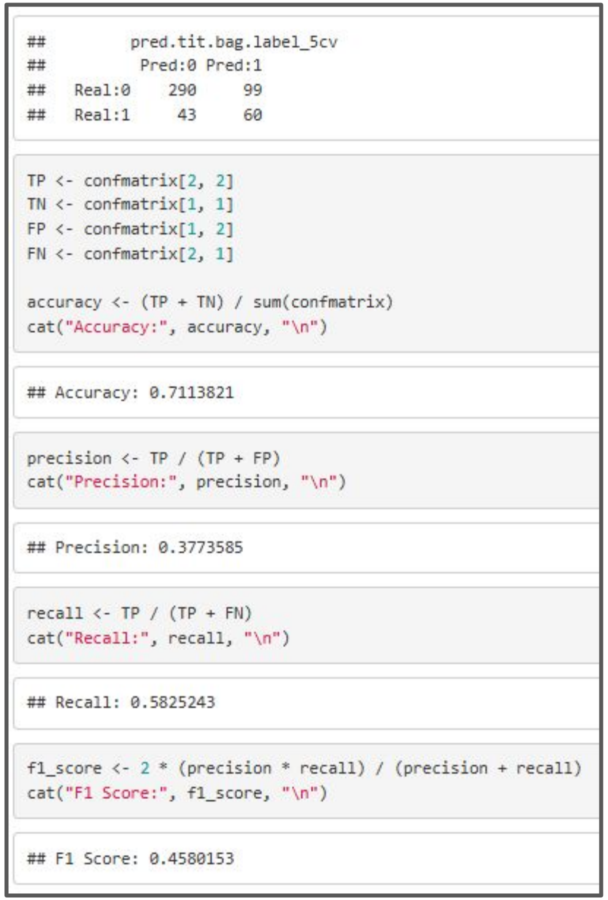
Modelo RF (re-entrenado sobre el nuevo train set)

```
##
## Call:
##  randomForest(formula = Status ~ ., data = train, ntree = 200,      mtry = mtry_cv, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 1230
##
##         OOB estimate of  error rate: 13.58%
## Confusion matrix:
##      0    1 class.error
## 0 1880  190  0.09178744
## 1  267 1029  0.20601852
```




QPC_Criteria_Order
QPC_date_spam
Joint_revision
QPC_Actual_Amount_real
Country.Code_744
Country_1
QPC_Adjusted_Amount_real
Country.Code_664
Country.Code_960
Country.Code_963
Country.Code_524
var_t1_t_NCG
Revised.Amount
correccion_T_NGDP_R
correccion_T_3_NCG
QPC_Original_Amount_BCA_t
correccion_T_3_D
var_t4_t_D
var_t1_t_D
shock_1
10
15
20
25
MeanDecreaseAccuracy
QPC_Criteria_Order
Joint_revision
QPC_date_spam
Country_1
QPC_Actual_Amount_real
Country.Code_744
QPC_Adjusted_Amount_real
Arrangement.Type_EFF
correccion_T_3_NCG
correccion_T_NGDP_R
var_t2_t1_BMG
correccion_T_3_D
correccion_T3_BMG
TX4_NM_gdp
var_t4_t_BXG
var_t4_t3_BMS
var_t4_t_BMS
var_t1_t_NCG
correccion_T4_PCPI
QPC_Original_Amount_BCA_t
0
20
40
60
80
100
120
140
MeanDecreaseGini

Performance de RF en test con threshold de validation



```
##        pred.tit.bag.label_5cv
##         Pred:0 Pred:1
##    Real:0    290     99
##    Real:1     43     60
```

```
TP <- confmatrix[2, 2]
TN <- confmatrix[1, 1]
FP <- confmatrix[1, 2]
FN <- confmatrix[2, 1]

accuracy <- (TP + TN) / sum(confmatrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.7113821
```

```
precision <- TP / (TP + FP)
cat("Precision:", precision, "\n")
```

```
## Precision: 0.3773585
```

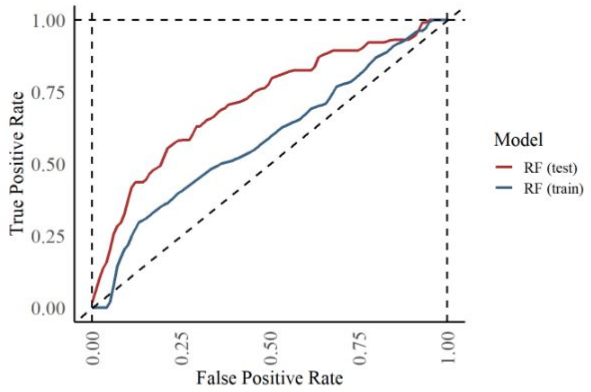
```
recall <- TP / (TP + FN)
cat("Recall:", recall, "\n")
```

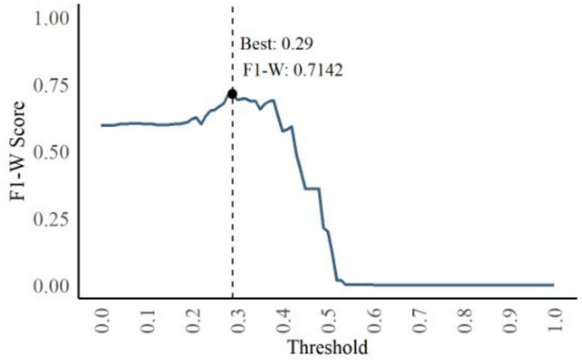
```
## Recall: 0.5825243
```

```
f1_score <- 2 * (precision * recall) / (precision + recall)
cat("F1 Score:", f1_score, "\n")
```

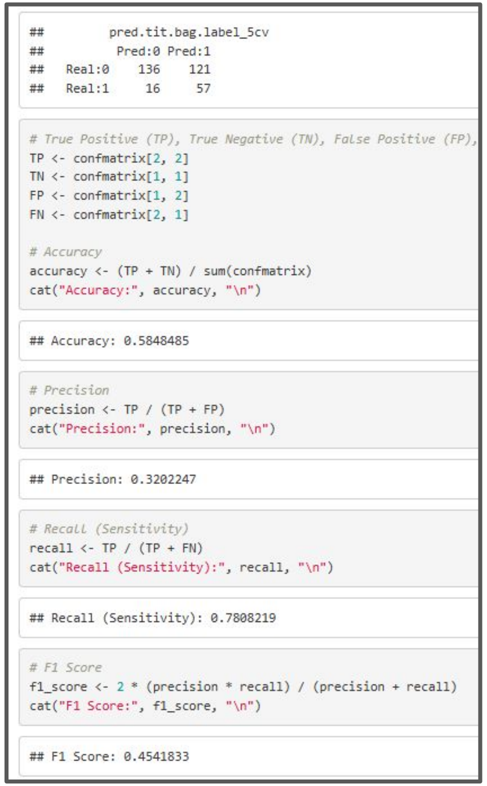
```
## F1 Score: 0.4580153
```

```
## Weighted F1 Score: 0.6873907
```





RF performance prediciendo QPCs fiscales (test set)



```
##          pred.tit.bag.label_5cv
##           Pred:0 Pred:1
##   Real:0    136    121
##   Real:1     16     57
```

```
# True Positive (TP), True Negative (TN), False Positive (FP),
TP <- confmatrix[2, 2]
TN <- confmatrix[1, 1]
FP <- confmatrix[1, 2]
FN <- confmatrix[2, 1]

# Accuracy
accuracy <- (TP + TN) / sum(confmatrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.5848485
```

```
# Precision
precision <- TP / (TP + FP)
cat("Precision:", precision, "\n")
```

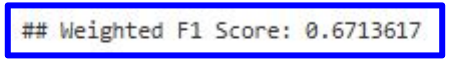
```
## Precision: 0.3202247
```

```
# Recall (Sensitivity)
recall <- TP / (TP + FN)
cat("Recall (Sensitivity):", recall, "\n")
```

```
## Recall (Sensitivity): 0.7808219
```

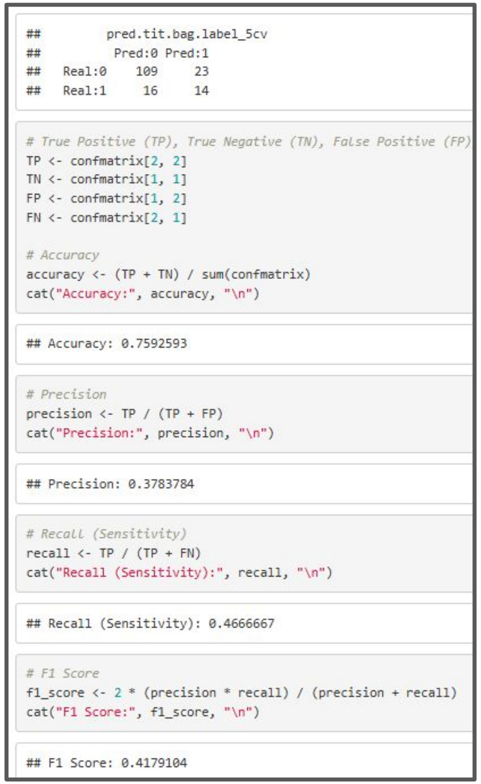
```
# F1 Score
f1_score <- 2 * (precision * recall) / (precision + recall)
cat("F1 Score:", f1_score, "\n")
```

```
## F1 Score: 0.4541833
```



```
## Weighted F1 Score: 0.6713617
```

RF performance prediciendo QPCs externos (test set)



```
##        pred.tit.bag.label_5cv
##          Pred:0 Pred:1
##   Real:0    109     23
##   Real:1     16     14
```

```
# True Positive (TP), True Negative (TN), False Positive (FP)
TP <- confmatrix[2, 2]
TN <- confmatrix[1, 1]
FP <- confmatrix[1, 2]
FN <- confmatrix[2, 1]

# Accuracy
accuracy <- (TP + TN) / sum(confmatrix)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.7592593
```

```
# Precision
precision <- TP / (TP + FP)
cat("Precision:", precision, "\n")
```

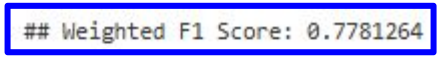
```
## Precision: 0.3783784
```

```
# Recall (Sensitivity)
recall <- TP / (TP + FN)
cat("Recall (Sensitivity):", recall, "\n")
```

```
## Recall (Sensitivity): 0.4666667
```

```
# F1 Score
f1_score <- 2 * (precision * recall) / (precision + recall)
cat("F1 Score:", f1_score, "\n")
```

```
## F1 Score: 0.4179104
```



```
## Weighted F1 Score: 0.7781264
```

RF performance prediciendo QPCs de principales deudores del FMI (test set)

Argentina

```
## Weighted F1 Score: 0.8235294
```

```
       pred.tit.bag.label_5cv
        Pred:0 Pred:1
Real:0       4      5
Real:1       1     14
```

Ucrania

```
Weighted F1 Score: 0.4000000
```

Egipto

```
Weighted F1 Score: 0.6000000
```

Pakistan

```
Weighted F1 Score: 0.7388462
```

Ecuador

```
Weighted F1 Score: 0.1463755
```

Respuestas de investigación

1. El modelo Random Forest performa mejor (out of sample) que el modelo benchmark: F1W = **0.69 versus 0.59,** respectivamente.

2. RF predice mejor metas **externas que fiscales:** F1W = **0.78 vs. 0.67** respectivamente.

3. El desempeño predictivo del modelo de ML para el caso de los principales deudores del FMI es mixto en términos de F1W: **Argentina (0.82), Ucrania (0.40), Egipto (0.60), Pakistán (0.74) y Ecuador (0.15).**