1. ETL Extract Transform Load

* Obtener los datos en un Excel y analizar cuáles sirven, depurar valores, ver si hay algo extra para calcular a priori.
  + Preguntas sobre los valores de los datos:
    - Se cambiaron los -- y vacíos por “no”, donde correspondía
    - Se normalizaron todos los valores de Plan 1988 y 88 a 88, ya que son equivalentes.
    - Los valores de Plan S95 y 1995 non son equivalentes.
    - Se marcaron como “usables = 0” los registros incompletos para no usarlos hasta tener la información necesaria. Faltan datos de Plan y adeuda trab final con ?
    - Una vez realizados estos cambios, se cambiaron los valores de sí y no por si=1 y no=0, para utilizar booleanos en la BD.
  + Se agregó el dato de Alumnos Recibido al final del 1er año de Reválidas (SI/NO)
* Importar los datos a la BD a partir de un Excel
  + Definir DBMS: PostgreSQL- listo
  + Copy con archivo CSV: cambiar permisos de lectura en el disco o mover el archivo a donde el usuario pueda abrirlo.

COPY public.prueba FROM 'D:\prueba.csv' USING DELIMITERS ';' with csv

* Agregar Store Procedures si es necesario calcular datos derivados a partir de las variables originales

1.2 Variables Involucradas

2. Calcular la Clasificación de Base para ver la precisión mínima a alcanzar

Total de alumnos inscriptos=123

Graduados al finalizar R1=50

Clasificación de Base = 40,65% de presición

3. Calcular con los distintos métodos sobre las distintas variables para ver cuáles podrían ser índices viables (según la precisión). Incluyendo el dato “Alumno Recibido” (S/N).

* OneR

**Script**

# query the data from postgreSQL

df\_postgres <- dbGetQuery(con, "SELECT anio\_ingreso,finales\_adeudados,cursadas\_adeudadas,optativas\_adeudadas,adeuda\_trab\_final\_inicio,plan,tutor\_r1, recibido\_fin\_r1 from revalidas WHERE usable=TRUE")

#df\_postgres <- dbGetQuery(con, "SELECT finales\_adeudados,cursadas\_adeudadas,optativas\_adeudadas,adeuda\_trab\_final\_inicio,plan,tutor\_r1, recibido\_fin\_r1 from revalidas WHERE usable=TRUE")

df\_postgres [ , ]

str(df\_postgres)

#install.packages(OneR)

require("OneR")

model <- OneR(df\_postgres, verbose = TRUE)

summary(model)

plot(model)

prediction <- predict(model, df\_postgres)

eval\_model(prediction, df\_postgres)

**Resultados grales:**

Attribute Accuracy

1 \* adeuda\_trab\_final\_inicio 62.6%

2 finales\_adeudados 60.98%

3 plan 59.35%

4 cursadas\_adeudadas 56.1%

4 optativas\_adeudadas 56.1%

4 tutor\_r1 56.1%

[podriamos agregar un one rule con una formula -ind ponderado x sentido común- ]

* J48

Script

## install.packages("RWeka")

require("RWeka")

df\_postgres <- dbGetQuery(con, "SELECT finales\_adeudados, cursadas\_adeudadas, optativas\_adeudadas, adeuda\_trab\_final\_inicio , plan, tutor\_r1, tutor\_r2, recibido\_fin\_r1 from revalidas")

## Identify a decision tree.

df\_postgres <- lapply(df\_postgres, as.factor)

m <- J48(recibido\_fin\_r1 ~., data = df\_postgres, control= Weka\_control(M=1,U=TRUE))

m

## Use 10 fold cross-validation.

e <- evaluate\_Weka\_classifier(m, cost = matrix(c(0,2,1,0), ncol = 2),numFolds = 10, complexity = TRUE,seed = 123, class = TRUE)

e

summary(e)

e$details

**Resultados:**

J48 unpruned tree

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

finales\_adeudados = 0

| plan = 1995

| | adeuda\_trab\_final\_inicio = FALSE

| | | optativas\_adeudadas = 0: TRUE (3.0)

| | | optativas\_adeudadas = 1: FALSE (1.0)

| | | optativas\_adeudadas = 2: TRUE (0.0)

| | | optativas\_adeudadas = 3: TRUE (0.0)

| | | optativas\_adeudadas = 4: TRUE (0.0)

| | | optativas\_adeudadas = 5: TRUE (0.0)

| | adeuda\_trab\_final\_inicio = TRUE: FALSE (3.0)

| plan = 88: FALSE (8.0/1.0)

| plan = S95

| | adeuda\_trab\_final\_inicio = FALSE: TRUE (7.0)

| | adeuda\_trab\_final\_inicio = TRUE

| | | tutor\_r1 = FALSE: TRUE (30.0/14.0)

| | | tutor\_r1 = TRUE

| | | | optativas\_adeudadas = 0: TRUE (26.0/9.0)

| | | | optativas\_adeudadas = 1: FALSE (1.0)

| | | | optativas\_adeudadas = 2: FALSE (2.0/1.0)

| | | | optativas\_adeudadas = 3: TRUE (0.0)

| | | | optativas\_adeudadas = 4: TRUE (0.0)

| | | | optativas\_adeudadas = 5: TRUE (0.0)

| plan = S96: TRUE (2.0)

| plan = S97: TRUE (1.0)

| plan = S98: TRUE (1.0)

finales\_adeudados = 1

| cursadas\_adeudadas = 0: FALSE (5.0)

| cursadas\_adeudadas = 1

| | plan = 1995: FALSE (0.0)

| | plan = 88: FALSE (2.0)

| | plan = S95

| | | optativas\_adeudadas = 0

| | | | adeuda\_trab\_final\_inicio = FALSE: TRUE (1.0)

| | | | adeuda\_trab\_final\_inicio = TRUE: FALSE (3.0/1.0)

| | | optativas\_adeudadas = 1: FALSE (1.0)

| | | optativas\_adeudadas = 2: FALSE (1.0)

| | | optativas\_adeudadas = 3: FALSE (0.0)

| | | optativas\_adeudadas = 4: FALSE (0.0)

| | | optativas\_adeudadas = 5: FALSE (0.0)

| | plan = S96: FALSE (0.0)

| | plan = S97: FALSE (0.0)

| | plan = S98: FALSE (0.0)

| cursadas\_adeudadas = 2: FALSE (0.0)

| cursadas\_adeudadas = 4: FALSE (0.0)

finales\_adeudados = 2

| tutor\_r1 = FALSE

| | adeuda\_trab\_final\_inicio = FALSE: TRUE (1.0)

| | adeuda\_trab\_final\_inicio = TRUE

| | | optativas\_adeudadas = 0: TRUE (1.0)

| | | optativas\_adeudadas = 1: FALSE (2.0)

| | | optativas\_adeudadas = 2: FALSE (1.0)

| | | optativas\_adeudadas = 3: FALSE (1.0)

| | | optativas\_adeudadas = 4: FALSE (0.0)

| | | optativas\_adeudadas = 5: FALSE (0.0)

| tutor\_r1 = TRUE: FALSE (8.0)

finales\_adeudados = 3: FALSE (6.0)

finales\_adeudados = 4: FALSE (5.0)

finales\_adeudados = 5: TRUE (1.0)

Number of Leaves : 45

Size of the tree : 59

>

> ## Use 10 fold cross-validation.

> e <- evaluate\_Weka\_classifier(m, cost = matrix(c(0,2,1,0), ncol = 2),numFolds = 10, complexity = TRUE,seed = 123, class = TRUE)

> e

=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances 77 62.0968 %

Incorrectly Classified Instances 47 37.9032 %

Kappa statistic 0.2466

Total Cost 65

Average Cost 0.5242

K&B Relative Info Score 3119.286 %

K&B Information Score 30.8527 bits 0.2488 bits/instance

Class complexity | order 0 122.5487 bits 0.9883 bits/instance

Class complexity | scheme 11887.7575 bits 95.869 bits/instance

Complexity improvement (Sf) -11765.2089 bits -94.8807 bits/instance

Mean absolute error 0.3665

Root mean squared error 0.4876

Relative absolute error 74.4985 %

Root relative squared error 98.3288 %

Total Number of Instances 124

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.586 0.333 0.695 0.586 0.636 0.251 0.686 0.743 FALSE

0.667 0.414 0.554 0.667 0.605 0.251 0.686 0.582 TRUE

Weighted Avg. 0.621 0.369 0.633 0.621 0.622 0.251 0.686 0.673

=== Cost Matrix ===

0 1

2 0

=== Confusion Matrix ===

a b <-- classified as

41 29 | a = FALSE

18 36 | b = TRUE

> summary(e)

Length Class Mode

string 1 -none- character

details 8 -none- numeric

detailsCost 1 -none- numeric

detailsComplexity 4 -none- numeric

detailsClass 12 -none- numeric

confusionMatrix 4 -none- numeric

> e$details

pctCorrect pctIncorrect pctUnclassified kappa meanAbsoluteError rootMeanSquaredError

62.0967742 37.9032258 0.0000000 0.2466391 0.3664706 0.4876487

relativeAbsoluteError rootRelativeSquaredError

74.4985114 98.3288003

[pegar resultados con las variables como estan]

[armar atributos derivados para rangos en las var de finales, optativas y cursadas adeudados y calcular con eso]

* Otros algoritmos…?

Analizar los resultados para cada combinación método-variable/s (anotar).

4. Automatizar el proceso de carga de datos en BD, cálculo del/los indicador/es y presentación de resultados.