# 

# Índice

[**Índice**](#_ngxwj72uif2o) **1**

[**Elaboración de un Indicador para el Programa de Reválidas**](#_49p4ou8e0hr) **2**

[Introducción](#_ha8u9ebt9kdj) 2

[Resumen del Procedimiento](#_7gq41su0png7) 2

[Algoritmos Evaluados](#_dvngywy80d29) 3

[OneR](#_9rr89znm22hk) 3

[Pseudocódigo](#_z4be1uyh38h4) 3

[Ejemplo](#_259z8ugv12s) 4

[Referencias](#_cphsjoovgxo7) 5

[J48](#_oi5lhlubyz3w) 5

[Ejemplo](#_yot1hbi5xp45) 5

[Referencias](#_3jb9w580rciv) 5

[Resultados](#_a1hwdngsvc8p) 6

[Atributos incluidos para ambos métodos](#_45t8h4sozdpv) 6

[OneR](#_641ffz5udeon) 6

[J48](#_3r496zymz81x) 8

[Árbol de decisión (sin podar)](#_1qc2lfr4e1lg) 8

[Breve análisis inicial](#_egkkkqht4vi7) 10

# 

# Elaboración de un Indicador para el Programa de Reválidas

## Introducción

En el problema actual, dados los valores de actividades académicas adeudadas y plan en el que se ha inscripto en la carrera de Ingeniería de Sistemas, se desea predecir basado en la experiencia previa si un alumno que se inscribe en el Plan de Reválidas será capaz de recibirse en los dos (2) años que dura el plan de reválidas automáticas.

## Resumen del Procedimiento

* Se corrigieron los valores de los datos recolectados para unificarlos, usando el procedimiento ETL (Extract, Transform and Load).
* Se creó una Base de Datos en PostgreSQL y se insertaron los datos recolectados para poder ejecutar comandos de R[[1]](#footnote-1) y realizar un análisis estadístico de los mismos.
* Se instaló y configuró CRan en Windows para correr comandos de R desde una consola contra PostgreSQL. Se instalaron los paquetes “OneR” y “RWeka” lo cual permitió correr los distintos algoritmos a evaluar.
* Se calculó la Clasificación de Base para ver la precisión mínima a alcanzar, ésta fue de 40,65%.
* Se investigaron dos (2) algoritmos de clasificación de datos sugeridos por la cátedra a partir de un trabajo previo con datos similares: One Rule y J48, utilizados para evaluar los métodos de rankeo de alumnos postulados para el programa Delta G.
* Se corrieron los algoritmos OneR y J48 para los mismos datos de alumnos inscriptos a R1 (Dic 2013-Dic2015, período finalizado) y R2 (Dic 2015 a Dic 2017, período no finalizado), con el objetivo de elaborar un indicador del estado del alumno que se inscribe al plan de reválidas respecto a la proyección de recibirse en el plazo de 2 años.

## 

## 

## Algoritmos Evaluados

### OneR

OneR viene de One Rule, es un algoritmo de clasificación que genera un árbol de decisión de un único nivel. OneR es capaz de inferir reglas de clasificación a partir de un conjunto de instancias. El algoritmo crea una regla para cada atributo en los datos de entrenamiento, luego escoge la regla con la *tasa de error*[[2]](#footnote-2) más pequeño como su "one rule". Para crear una regla para cada atributo debe determinarse la clase más frecuente para cada valor del atributo.

Las desventajas que podemos advertir en este algoritmo son:

i) El algoritmo trata todos los atributos numéricamente evaluados como continuos, usa un método directo para dividir el rango de valores en intervalos disjuntos. Esto introduce un riesgo de "overfitting" en el caso de atributos evaluados de forma continua, por ejemplo: números de teléfono, etc…

ii) El "overfitting" de atributos nominales con valores únicos tales como nombres de personas, direcciones de correo electrónico, etc…

iii) Selección aleatoria de un atributo cuando las tasas de error son iguales.

iv) Selección aleatoria de una clase cuando dos o más clases dan la misma tasa de error con un atributo.

**Overfitting** **(o sobreajuste)**: es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado. cuando un sistema se entrena demasiado (se sobreentrena) o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo.

**Accuracy**: porcentaje de precisión de la predicción.

#### Pseudocódigo

Para cada atributo o “predictor”[[3]](#footnote-3)

Para cada valor del atributo, crear una regla como sigue:

Contar las apariciones de cada valor del target[[4]](#footnote-4) (atributo a predecir: recibido)

Seleccionar la clase[[5]](#footnote-5) más frecuente

Hacer que la regla asigne esa clase a este valor del atributo o “predictor”

Calcular el error total de las reglas de cada atributo predictor

Elegir el atributo “predictor” con el menor error total

#### Ejemplo

Al correr el algoritmo OneR con los datos recolectados, se calcula la precisión de cada atributo y en base a eso, se realiza la selección del atributo a partir del cuál se van a generar las reglas.

A continuación se muestra, a modo de ejemplo, el análisis realizado para los dos atributos con más precisión dentro del conjunto de atributos (para ver todos los resultados ir más adelante, a la sección “Resultados”).

**Análisis de la precisión para el atributo adeuda\_trabajo\_final\_inicio**

|  |  |  |  |
| --- | --- | --- | --- |
| adeuda\_trab\_final\_inicio\recibido | 1 | 0 | Precisión |
| 1 | 39 | 63 | 0,6290322581 |
| 0 | 15 | 7 |

La precisión de este atributo es 0,6290322581

(63 + 15) / 124

**Análisis de la precisión para Finales adeudados**

|  |  |  |  |
| --- | --- | --- | --- |
| Finales\_adeudados\Recibido | 1 | 0 | Precisión |
| 0-1 | 51 | 47 | 0,6048387097 |
| 2 | 2 | 12 |
| 3 | 0 | 6 |
| 4 | 0 | 5 |
| 5 | 1 | 0 |

La precisión de este atributo es 0,6048387097

(51 + 12 + 6 + 5 + 1) / 124

**Criterio de selección**

El atributo adeuda\_trabajo\_final\_inicio tiene un accuracy mayor, 0,629..., que finales\_adeudados , 0,604. (Es lo mismo que decir que tiene menor tasa de fallo que finales\_adeudados .) Por lo que el algoritmo escoge adeuda\_trabajo\_final\_inicio.

Esto se hace con todos los atributos y se selecciona el que menor tasa de error tenga. (O mayor accuracy). En base al atributo seleccionado, se generan las reglas, basado en los valores que éste puede tomar.

Como se mencionó anteriormente, luego de comparar todos los atributos, se encuentra que el atributo con mejor porcentaje de precisión es **adeuda\_trabajo\_final** y las reglas generadas para este atributo son:

If adeuda\_trab\_final\_inicio = FALSE then recibido = TRUE

If adeuda\_trab\_final\_inicio = TRUE then recibido = FALSE

#### Referencias

* http://www.saedsayad.com/oner.htm

### J48

Este algoritmo construye árboles de decisión desde un grupo de datos de entrenamiento usando el concepto de [entropía de](https://es.wikipedia.org/w/index.php?title=Entropia_(teor%C3%ADa_de_informaci%C3%B3n)&action=edit&redlink=1) información. En cada nodo del árbol, el algoritmo elige el atributo de los datos que más eficazmente divida el conjunto de muestras en subconjuntos enriquecidos en una clase u otra. Su criterio es el normalizado para ganancia de información (diferencia de entropía) que resulta en la elección de un atributo para dividir los datos. El atributo con la mayor ganancia de información normalizada se elige como parámetro de decisión. El algoritmo divide el conjunto de datos recursivamente en sublistas más pequeñas.

**Entropía**: se puede considerar como la cantidad de información promedio que contienen los símbolos usados. Los símbolos con menor probabilidad son los que aportan mayor información; por ejemplo, si se considera como sistema de símbolos a las palabras en un texto, palabras frecuentes como «que», «el», «a» aportan poca información, mientras que palabras menos frecuentes como «corren», «niño», «perro» aportan más información. Cuando todos los símbolos son igualmente probables (distribución de probabilidad plana), todos aportan información relevante y la entropía es máxima.

#### Ejemplo

Una rama -> finales\_adeudados = 0

| plan = 88: => FALSE (8.0/1.0)

Quiere decir que si no se adeudan finales y pertenece al plan 88, se predice que no se van a recibir y una sola predicción de cada 8 falla. (7/8 = 0,875, se predice con un 87,5% de acierto)

Como los otros atributos no suman información a esta rama, se corta ahí.

#### Referencias

* An Introduction to Data Mining, [Dr. Saed Sayad](http://www.saedsayad.com/author.htm):<http://www.saedsayad.com/decision_tree.htm>

Mineria De Datos En Weka, Francisco Ferrero Mateos: http://www.it.uc3m.es/~jvillena/irc/practicas/03-04/20.mem.pdf

## Resultados

### Atributos incluidos para ambos métodos

* finales\_adeudados
* cursadas\_adeudadas
* optativas\_adeudadas
* adeuda\_trab\_final\_inicio
* recibido

### OneR

Atributo Precisión

1 \* adeuda\_trab\_final\_inicio 62.9%

2 finales\_adeudados 60.48%

3 cursadas\_adeudadas 56.45%

3 optativas\_adeudadas 56.45%

**Atributo seleccionado dada la precisión:** '\*' (adeuda\_trabajo\_final\_inicio)

**Reglas:**

If adeuda\_trab\_final\_inicio = FALSE then recibido = TRUE

If adeuda\_trab\_final\_inicio = TRUE then recibido = FALSE

**Precisión:**

78 de124 instancias clasificadas correctamente (62.9%)

**Tabla de contingencia:**

|  |  |  |  |
| --- | --- | --- | --- |
| adeuda\_trab\_final\_inicio \ recibido | FALSE | TRUE | Sum |
| FALSE | 7\* | 63 | 70 |
| TRUE | 15\* | 39 | 54 |
| Sum | 22 | 102 | 124 |

**Máximo de cada columna:** '\*'

**Matriz de Confusión (absoluta):**

Actual FALSE TRUE Sum

Predicción

FALSE 63 39 102

TRUE 7 15 22

Sum 70 54 124

**Matriz de Confusión (relativa):**

Actual FALSE TRUE Sum

Predicción

FALSE 0.51 0.31 0.82

TRUE 0.06 0.12 0.18

Sum 0.56 0.44 1.00

**Precisión:**

0.629 (78/124)

**Tasa de Error:**

0.371 (46/124)

**Reducción de Tasa de Errores (vs. tasa base):**

0.1481 (p-value = 0.08651)

### 

### J48

#### Árbol de decisión (sin podar)

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: TRUE (70.0/25.0)

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)

| cursadas\_adeudadas = 2: FALSE (0.0)

| cursadas\_adeudadas = 4: FALSE (0.0)

finales\_adeudados = 2

| adeuda\_trab\_final\_inicio = FALSE

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

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

| | cursadas\_adeudadas = 2: FALSE (0.0)

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

| adeuda\_trab\_final\_inicio = TRUE: FALSE (12.0/1.0)

finales\_adeudados = 3: FALSE (6.0)

finales\_adeudados = 4: FALSE (5.0)

finales\_adeudados = 5: TRUE (1.0)

Número de hojas: 29

Tamaño del árbol : 39

Instancias correctamente clasificadas: 86 (69.3548 % de precisión)

Instancias incorrectamente clasificadas: 38 (30.6452 % de error)

=== Matriz de Confusión J48 ===

|  |  |  |
| --- | --- | --- |
| a | b | <-- clasificado como |
| 41 | 29 | a = FALSE |
| 9 | 45 | b = TRUE |

# 

## Breve análisis inicial

Mientras que OneR considera los atributos de forma aislada y genera reglas en base al atributo que más se correlaciona al valor “true” o “false” de recibido, J48 considera todos los atributos que brinden información de decisión para llegar a un valor de *“recibido”* y predecir con qué certeza se puede afirmar que el alumno se va a recibir o no.

[HACER] Comparar resultados de OneR vs J48 desde el punto de vista de los atributos que eligió primero uno y el otro: ejemplo adueda\_trabajo\_final\_inicio es tomado en OneR como la variable con mayor precisión, sin embargo esta misma aparece en el árbol generado por J48 en el nivel 3. Qué significa? Qué nos dice respecto de los datos? Y de los métodos?

Creemos que J48 representa mejor la situación de un alumno que se inscribe en el programa de reválidas, ya que no es suficiente saber cuál es el factor que más influye al momento de recibirse, sino que en este caso nos parece que varios factores -tal vez de menor influencia- conjuntamente pueden afectar al resultado final de igual manera que un factor solo de mayor influencia.

Es decir, un alumno que al comenzar el plan de reválidas adeuda 5 finales, pero hace mucho que no rinde nada, quizá tenga la misma posibilidad de recibirse que un alumno que adeuda el trabajo final únicamente y se encuentra en ritmo de estudio (su última actividad académica[[6]](#footnote-6) es reciente), y tal vez otro que adeude cursadas y finales tenga las mismas chances. Por esto, pensamos que un método que evalúe todas las alternativas es mejor que uno que genere reglas por cada atributo o dato a considerar.

Por otro lado, se podría realizar OneR con combinaciones de valores o crear un índice que sea una fórmula ponderada, pero sería un método ad-hoc que debería ajustarse de acuerdo a la correlación de los valores predichos, con los valores reales al final del plan R2. Dicho de otro modo, verificando qué correlación hay entre los alumnos que se predijo que se iban a recibir y los que efectivamente lo hicieron. [HACER: quizá valga la pena hacer pruebas y comprobar o no que, efectivamente, no es tan bueno?]

Una limitación que vemos al momento de definir un método para obtener el/los indicadores, es que los datos de R1 están “completos” porque el período del ciclo ya terminó, mientras que los de R2 están incompletos, dado que al finalizar el período es posible que haya más alumnos recibidos que al momento de recolectar estos datos. Esto significa que, en el peor de los casos, la precisión va a mejorar porque van a existir mas datos de entrenamiento y serán más precisos sobre la situación final de los alumnos.óe

1. **R** (<https://www.r-project.org/>) es un entorno de software libre para informática estadística y gráficos. Se compila y se ejecuta en una amplia variedad de plataformas/sistemas operativos. [↑](#footnote-ref-1)
2. Tasa de error: número de instancias de los datos de entrenamiento en los que la clase del valor de un atributo no concuerda con la asociación que la regla le da al valor de ese atributo. [↑](#footnote-ref-2)
3. Atributos/predictores: son los datos de entrada que ayudarán a predecir el target,en nuestro caso cant\_finales\_adeudados, adeuda\_trabajo\_final\_inicio, cant\_cursadas\_adeudadas, etc. [↑](#footnote-ref-3)
4. Target: es el atributo a predecir, en nuestro caso “recibido”. [↑](#footnote-ref-4)
5. Clase: es la cantidad de veces que aparece cada valor de un atributo. [↑](#footnote-ref-5)
6. Pregunta para Mariana y Laura: Tenemos datos sobre la ultima actividad académica de los alumnos al momento de inscribirse a las reválidas? De no ser así, podemos plantear una mejora a la generación de índices de incluir este dato. [↑](#footnote-ref-6)