

Inteligencia en redes de comunicaciones

Practica WEKA

Diagnóstico cardiología

Alejandro Boris Valiente NIA 100029722
Zaida Cebrián Jiménez NIA 100029587

1. Introducción	3
1.1. <i>Introducción al problema. Arritmias cardiacas</i>	3
1.2. <i>Introducción a Minería de datos</i>	3
2. Weka	4
3. Preprocesado de datos	5
3.1. <i>Datos</i>	5
3.2. <i>Preprocesamiento</i>	5
4. Procesado de datos	6
4.1. Algoritmos utilizados	7
4.1.1. <i>Algoritmo OneR</i>	7
4.1.2. <i>Algoritmo J48</i>	7
4.1.3. <i>Algoritmo SMO</i>	7
4.1.4. <i>Algoritmo IBk</i>	8
4.2. Resultados	8
4.2.1. <i>Test A: Clasificación en pacientes N y T</i>	8
4.2.1.1. Algoritmo OneR	9
4.2.1.1.1. <i>Entrenamiento de la máquina</i>	9
4.2.1.1.2. <i>Clasificación pacientes</i>	10
4.2.1.2. Algoritmo J48	11
4.2.1.2.1. <i>Entrenamiento de la máquina</i>	11
4.2.1.2.2. <i>Clasificación pacientes</i>	12
4.2.1.3. Algoritmo IB1	12
4.2.1.3.1. <i>Entrenamiento de la máquina</i>	12
4.2.1.3.2. <i>Clasificación pacientes</i>	14
4.2.1.4. Algoritmo IBk	14
4.2.1.4.1. <i>Entrenamiento de la máquina</i>	14
4.2.1.4.2. <i>Clasificación pacientes</i>	15
4.2.1.5. Resultados finales	16
4.2.2. <i>Test B: Clasificación en pacientes S y T</i>	16
4.2.2.1. Algoritmo OneR	17
4.2.2.1.1. <i>Entrenamiento de la máquina</i>	18
4.2.2.1.2. <i>Clasificación pacientes</i>	18
4.2.2.2. Algoritmo SMO	18
4.2.2.2.1. <i>Entrenamiento de la máquina</i>	18
4.2.2.2.2. <i>Clasificación pacientes</i>	19
4.2.2.3. Algoritmo IB5	20
4.2.2.3.1. <i>Entrenamiento de la máquina</i>	20
4.2.2.3.2. <i>Clasificación pacientes</i>	21
4.2.2.4. Resultados finales	21
5. Conclusiones	22
6. Bibliografía	22
7. Anexo A: Tabla de resultados de la clasificación del test A	23
8. Anexo B: Tabla de resultados de la clasificación del test B	30

1. Introducción

En esta práctica se van a analizar a una serie de pacientes de arritmias cardíacas a través de técnicas de minería de datos. Se analizará que algoritmo es más conveniente para nuestro problema, para luego hacer una clasificación de todos los pacientes. Además, se tratará en profundidad el preprocesado de datos que se tuvo que hacer antes así como el criterio seguido para hacerlo.

Antes de seguir con los siguientes apartados, se verá una breve introducción a las dos principales partes sobre las que versa la práctica: Minería de datos y arritmias cardíacas, qué son y en qué consiste cada una.

1.1. Introducción al problema. Arritmias cardíacas

La *fibrilación atrial* está considerada como la arritmia cardíaca más común en los seres humanos, afectando a más de 2 millones de personas sólo en EEUU. A diferencia de la fibrilación ventricular, que es letal si no se interrumpe, es posible mantener una *fibrilación atrial* (auricular) indefinidamente, puesto que los ventrículos pueden suplir parcialmente la labor de bombeo de las aurículas. Sin embargo, es una enfermedad muy seria puesto que a la larga puede producir ataques de corazón e infartos de miocardio causados por la formación de coágulos en sangre.

Lo que se pretende conseguir con la minería de datos es predecir la terminación de estas arritmias. Para ello, se cuenta con una serie de registros ECG (electrocardiogramas) obtenidos de pacientes arritmicos. A partir del estudio de estas señales y de su clasificación, se podrá llegar a predecir la terminación de las mismas.

La idea es que si se logra entender y predecir cuándo se van a producir los episodios de terminación espontánea de AF, esto puede dar lugar a mejoras en el tratamiento de la AF crónica y, por tanto, mejoras en la calidad de vida de los pacientes.

Como se puede observar, la minería de datos puede aplicarse entre otras muchas cosas, a la medicina, enfocándose en la mejora en la calidad de vida de los pacientes, ya sea de arritmias cardíacas o de cualquier otra enfermedad de la que se puedan extraer el suficiente número de datos.

1.2. Introducción a Minería de datos

En los últimos años está aumentando la relevancia de las tecnologías basadas en procesamiento de información, entre ellas la minería de datos. Así, lo que en un principio estaba limitado a pequeñas áreas, se ha extendido a todos los ámbitos de la sociedad donde pueda tener utilidad.

La minería de datos es un proceso analítico diseñado para explorar grandes volúmenes de datos con el objeto de descubrir patrones y modelos de comportamiento o relaciones entre diferentes variables. Por esto, la minería de datos se utiliza como herramienta de análisis y descubrimiento de conocimiento a partir de datos de observación o de resultados de experimentos.

El proceso de la minería de datos se puede dividir en las siguientes fases:

- *Filtrado de datos:* El formato de los datos contenidos en las bases de datos nunca es el idóneo, por ello, mediante el preprocesado, se filtran los datos según las necesidades y el algoritmo a usar, se obtienen muestras de los mismos o se reducen el número de valores posibles.
- *Selección de variables:* Aún después de haber sido preprocesados, en la mayoría de los casos se tiene una cantidad ingente de datos. es necesario seleccionar determinadas características influyentes en el problema para reducir el volumen de datos obtenido anteriormente.
- *Extracción de conocimiento:* Mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables
- *Interpretación y evaluación:* Una vez obtenido el modelo, se debe proceder a su validación, comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias.

2. Weka

En la presente práctica se van a desarrollar tanto el aprendizaje por parte de la máquina como la verificación de lo aprendido, por medio de tests, mediante la herramienta *WEKA*.

WEKA es una herramienta que nos permite realizar minería de datos con una interfaz gráfica para facilitar su utilización. Además, permite una comparación con los distintos métodos que se utilizan para el pre-procesamiento, clasificación de información, clustering y meta-aprendizaje.

3. Preprocesado de datos

3.1. Datos

Se dispone de 80 registros de ECG en pacientes afectados por AF, cada uno de un minuto de duración y conteniendo dos señales ECG muestreadas a 128 muestras por segundo.

El conjunto de entrenamiento contiene 30 registros en total, con 10 registros por cada uno de los tres grupos siguientes:

- Grupo N (registros n01, n02, ... n10): AF no-finalizada (AF que no finalizó en la hora siguiente al registro ECG del paciente)
- Grupo S (registros s01, s02, ... s10): AF que termina un minuto tras el final del registro ECG.
- Grupo T (registros t01, t02, ... t10): AF que termina inmediatamente a continuación del registro ECG (en 1 segundo). Obsérvese que estos registros siguen de forma inmediata en el tiempo a los registros del Grupo S (es decir, t01 es la continuación de s01).

El conjunto de entrenamiento fue obtenido de 20 sujetos diferentes (10 grupo N, 10 grupo S/T).

El conjunto de prueba A contiene 30 registros (a01, a02, ... a30) de 30 sujetos (ninguno contenido en el conjunto de entrenamiento o en el conjunto de prueba B). La mitad de ellos (aprox.) pertenecen al grupo N y los restantes al T. el objetivo es identificar cuáles de ellos son del grupo T.

El conjunto de prueba B contiene 20 registros (b01, b02, ... b20), cada 2 del mismo sujeto (ninguno en conjunto de entrenamiento ni en conjunto de prueba A). Un registro del par es del grupo S y otro del T. El objetivo es identificar qué registros pertenecen al grupo T.

3.2. Preprocesamiento

Para entrenar a la máquina y clasificar a continuación las muestras de los tests es necesario crear los siguientes 4 ficheros arff:

- *nt.arff*: Set de entrenamiento con los registros de los pacientes de tipo N y tipo T preprocesados como se indicó anteriormente.
- *st.arff*: Set de entrenamiento con los registros de los pacientes de tipo S y tipo T preprocesados como se indicó anteriormente.
- *test_a.arff*: Consta de los registro clasificar en pacientes clase N y clase T.
- *test_b.arff*: Consta de los registro clasificar en pacientes clase S y clase T.

Para la creación de estos ficheros se siguió siempre el mismo criterio: reducir el número de muestras de los dos registros del paciente y desglosar ambos registros en más, con la finalidad de tener más casos con los que entrenar a la máquina.

La idea principal para su procesamiento radicaba en que la señal tenía dependencia temporal, es decir, que no daba lo mismo tomar valores del principio que del final. Como se nos pedía diferenciar las señales que duraban una hora de las que duraban uno y dos minutos, se decidió tomar más muestras de los últimos 10 segundos de la señal que del resto, dándole así más importancia al final de la señal.

Se decidió obtener de cada registro 5 señales distintas y características de la clase de arritmia en cuestión con la finalidad de no entrenar a la máquina siempre para el mismo conjunto de valores. La estructura de las subseñales usadas es la siguiente:

```
[x(1) x(6) x(11) x(16) x(21) x(26) x(31) x(36) x(41) x(46) x(50.5) x(53.0) x(55.5) x(58.0)]  
[x(2) x(7) x(12) x(17) x(22) x(27) x(32) x(37) x(42) x(47) x(51.0) x(53.5) x(55.5) x(58.5)]  
[x(3) x(8) x(13) x(18) x(23) x(28) x(33) x(38) x(43) x(48) x(51.5) x(54.0) x(55.5) x(59.0)]  
[x(4) x(9) x(14) x(19) x(24) x(29) x(34) x(39) x(44) x(49) x(52.0) x(54.5) x(55.5) x(59.5)]  
[x(5) x(10) x(15) x(20) x(25) x(30) x(35) x(40) x(45) x(50) x(52.5) x(55.0) x(55.5) x(60.0)]
```

Siendo x la señal ECG, y siendo el número entre paréntesis el tiempo en el que se encuentra la muestra. Los valores de la señal muestreada se podían obtener por medio de la herramienta *Chart-O-Matic*, incluida en la página web del proyecto de cardiología sobre el que versa esta práctica.

Así, finalmente, y tras seguir el método anteriormente explicado, se consiguieron 10 registros distintos por paciente, con 14 atributos numéricos más uno nominal (la clase de arritmia que padece), en cada uno.

4. Procesado de datos

Finalmente, y una vez realizado todo el preprocesamiento de los datos proporcionados, se pasó al procesamiento de los mismos para clasificar correctamente los pacientes de los tests.

Lo que se busca es aquel algoritmo que de los mejores resultados de clasificación una vez entrenada la máquina. Para ellos se probará con un determinado número de algoritmos incluidos en la herramienta WEKA (no todos ya que existen algunos que ni siquiera funciona con valores nominales como es nuestro caso) y a partir de los resultados ofrecidos se determinará cual es la mejor clasificación.

Para ello, serán analizados en una primera instancia los algoritmos utilizados para más adelante observar los resultados que ellos mismos producen.

4.1. Algoritmos utilizados

Los algoritmos que se usarán en esta práctica para alcanzar los objetivos marcados se muestran a continuación.

4.1.1. Algoritmo OneR

El algoritmo OneR (one rule) forma parte de los algoritmos basados en árboles de decisión, en los que el conocimiento obtenido en el proceso de aprendizaje se representa mediante un árbol. En él, cada nodo interior contiene una pregunta sobre un atributo concreto y cada hoja del árbol se refiere a una de las posibles clasificaciones.

Este algoritmo en concreto, está basado en un árbol de profundidad 1 (un solo nivel), usando por tanto, una única regla de decisión.

Lo que hará será definir reglas simples a partir de las instancias. Así, seleccionará el atributo con el menor error cuadrático medio; para crear una regla a partir de un atributo, se elegirá la clase más frecuente de dicho atributo, es decir, la muestra que aparece más a menudo para un valor del atributo.

En su implementación en WEKA, este algoritmo elige la regla con el número más alto de instancias correctas, no la menor tasa de error.

4.1.2. Algoritmo J48

Este algoritmo (implementación hecha por WEKA basado en el algoritmo conocido como c4.5), forma parte también de los algoritmos basados en árboles de decisión, al igual que el algoritmo anterior.

La característica fundamental de este algoritmo es que incorpora una poda del árbol de clasificación una vez que éste ha sido inducido, es decir, una vez construido el árbol de decisión, se podan aquellas ramas del árbol con menor capacidad predictiva.

Este algoritmo es una mejora de ID3, también basado en árboles, donde el criterio escogido para seleccionar la variable más informativa está basado en el concepto de cantidad de información mutua entre dicha variable y la variable clase.

4.1.3. Algoritmo SMO

Este algoritmo está basado en redes neuronales (funcionamiento inspirado en el cerebro humano, de ahí su nombre) cuya característica más importante es su capacidad de aprender a partir de ejemplos, lo cual les permite generalizar sin tener que formalizar el conocimiento adquirido.

El algoritmo SMO se caracterizará por tener un aprendizaje no supervisado competitivo y por no tener ningún resultado objetivo al que la red deba tender. Además, SMO divide el problema en una serie de problemas más pequeños que se resuelven de forma analítica.

Cada neurona de la red calcula la similitud entre el vector de entrada y su propio vector de pesos según un criterio de similitud establecido. A continuación, simulando un proceso competitivo, se declara vencedora la neurona cuyo vector de pesos es el más similar al de entrada. Esto, hace que la red SOM se comporte como un clasificador, ya que la neurona de salida activada representará la clase a la que pertenece la información de entrada.

4.1.4. Algoritmo IBk

Este algoritmo está basado en instancias, por ello consiste únicamente en almacenar los datos presentados. Cuando una nueva instancia es encontrada, un conjunto de instancias similares relacionadas es devuelto desde la memoria y usado para clasificar la instancia consultada.

Se trata, por tanto, de un algoritmo del método *lazy learning*. Este método de aprendizaje se basa en que los módulos de clasificación mantienen en memoria una selección de ejemplos sin crear ningún tipo de abstracción en forma de reglas o de árboles de decisión (de ahí su nombre, *lazy*, perezosos). Cada vez que una nueva instancia es encontrada, se calcula su relación con los ejemplos previamente guardados con el propósito de asignar un valor de la función objetivo para la nueva instancia.

La idea básica sobre la que se fundamenta este algoritmo es que un nuevo caso se va a clasificar en la clase más frecuente a la que pertenecen sus K vecinos más cercanos. De ahí que sea también conocido como método K-NN: *K Nearest Neighbours*.

4.2. Resultados

Después de ver todos los algoritmos que se van a utilizar en la presente práctica, a continuación se muestran los resultados obtenidos para cada uno de los algoritmos a partir de los datos preprocesados y el test correspondiente.

4.2.1. Test A: Clasificación en pacientes N y T

Como se dijo anteriormente, el objetivo es conseguir clasificar las muestras contenidas en este test en dos grupos de pacientes: N y T.

Para ello, la máquina será entrenada para los algoritmos anteriores con el fichero nt.arff creado en el preprocesado, para ser a continuación probada con el test correspondiente.

Lo ideal sería obtener unos resultados de clasificación del 50% de los datos de cada tipo. A continuación se muestran los resultados obtenidos.

4.2.1.1. Algoritmo OneR

4.2.1.1.1. Entrenamiento de la máquina

Tras entrenar la máquina con el fichero adecuado se obtiene el siguiente resumen. En él se muestra un 70% de instancias bien clasificadas, lo que arroja un posible error en evaluaciones posteriores muy grande.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      210          70      %
Incorrectly Classified Instances    90          30      %
Kappa statistic                    0.39
Mean absolute error                 0.305
Root mean squared error             0.5523
Relative absolute error             61      %
Root relative squared error         110.4536 %
Total Number of Instances          300

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
0.8         0.41         0.661       0.8       0.724        n
0.59        0.2          0.747       0.59      0.659        t

=== Confusion Matrix ===

  a  b  <-- classified as
120 28 | a = n
 62 90 | b = t
```

Una vez entrenado, comprobamos la validez del entrenamiento mediante la función split del WEKA, un porcentaje de las muestras será destinado al entrenamiento y el resto a la verificación. En este caso, el porcentaje de muestras dedicado al entrenamiento es el 80% como se recomendaba en el curso de Tratamiento Digital de Señales. El resultado se presenta a continuación.

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances      24          60      %
Incorrectly Classified Instances    16          40      %
Kappa statistic                    0.1899
Mean absolute error                 0.4
Root mean squared error             0.6325
Relative absolute error             79.9506 %
Root relative squared error         126.4035 %
Total Number of Instances          40
```

```
=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.474	0.286	0.6	0.474	0.529	n
0.714	0.526	0.6	0.714	0.652	t

```
=== Confusion Matrix ===
```

```

a  b  <-- classified as
9 10 |  a = n
6 15 |  b = t

```

4.2.1.1.2. Clasificación pacientes

A continuación, se procede a clasificar los pacientes del fichero test_a. Suponiendo a todos los pacientes n, estos son los resultados que arroja el procesamiento.

```
=== Evaluation on test set ===
```

```
=== Summary ===
```

Correctly Classified Instances	177	59	%
Incorrectly Classified Instances	123	41	%
Kappa statistic	0		
Mean absolute error	0.41		
Root mean squared error	0.6403		
Relative absolute error	82	%	
Root relative squared error	128.0625	%	
Total Number of Instances	300		

```
=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.59	0	1	0.59	0.742	n
0	0.41	0	0	0	t

```
=== Confusion Matrix ===
```

```

a  b  <-- classified as
177 123 |  a = n
0    0  |  b = t

```

Como se puede observar el 59% de las muestras las clasifica como n, mientras que el 41% restante como t. No es mal resultado si se le compara con el 50 % ideal, pero aún así y en vistas de lo obtenido en el entrenamiento cabe pensar que se habrá cometido algún error. Además, al solo evaluar un único atributo, es más probable que se cometan errores.

Hay que resaltar además, que el número de pacientes es en realidad 30, teniendo cada paciente 10 señales distintas que deciden la clase de arritmia que padece, por lo que puede que en el cómputo total la clasificación sea mejor o peor de lo que aquí aparece.

4.2.1.2. Algoritmo J48

4.2.1.2.1. Entrenamiento de la máquina

Entrenando a la máquina mediante el uso del algoritmo J48, el WEKA arroja el siguiente resumen de resultados.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      176           88      %
Incorrectly Classified Instances    24           12      %
Kappa statistic                    0.76
Mean absolute error                 0.1812
Root mean squared error             0.301
Relative absolute error             36.2417 %
Root relative squared error         60.2011 %
Total Number of Instances          200

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
    0.9      0.14      0.865      0.9      0.882      n
    0.86      0.1      0.896      0.86      0.878      t

=== Confusion Matrix ===

  a  b  <-- classified as
90 10 |  a = n
14 86 |  b = t
```

A diferencia del caso anterior, en éste el 88% de las muestras están bien clasificadas. Al igual que antes, se procede a entrenar con el 80% y a probar con el 20% para ver si en efecto la máquina está tan bien entrenada.

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances      21           52.5    %
Incorrectly Classified Instances    19           47.5    %
Kappa statistic                    0.064
Mean absolute error                 0.4544
Root mean squared error             0.5947
Relative absolute error             90.8228 %
Root relative squared error         118.8536 %
Total Number of Instances          40

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
    0.684      0.619      0.5      0.684      0.578      n
    0.381      0.316      0.571      0.381      0.457      t

=== Confusion Matrix ===

  a  b  <-- classified as
13  6 |  a = n
13  8 |  b = t
```

Los resultados obtenidos son si cabe peores que los que se obtuvieron en el anterior apartado. Llama la atención estos resultados sobre todo en la medida en que éste algoritmo tiene más atributos en cuenta que el anterior.

4.2.1.2.2. *Clasificación pacientes*

Por último clasificamos el test arrojando los resultados presentados a continuación.

```

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      166           55.3333 %
Incorrectly Classified Instances    134           44.6667 %
Kappa statistic                     0
Mean absolute error                 0.4788
Root mean squared error             0.6336
Relative absolute error             95.7602 %
Root relative squared error         126.7123 %
Total Number of Instances          300

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.553      0          1           0.553     0.712        n
  0          0.447      0           0         0           t

=== Confusion Matrix ===

  a    b    <-- classified as
166 134 |    a = n
  0    0 |    b = t

```

Se puede observar que el porcentaje de las instancias bien clasificadas se parece más al ideal que el alcanzado con el anterior algoritmo. Al igual que en la vez anterior, es mayor el número de instancias n asignadas que el de instancias t.

4.2.1.3. *Algoritmo IB1*

4.2.1.3.1. *Entrenamiento de la máquina*

Al igual que en los apartados anteriores, antes de la clasificación del fichero test hay que entrenar la máquina. El algoritmo IBk fue seleccionado con K=1, es decir, profundidad 1. Los resultados tras el entrenamiento se muestran a continuación.

```
=== Evaluation on training set ===
=== Summary ===
```

Correctly Classified Instances	200	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0		
Root mean squared error	0		
Relative absolute error	0	%	
Root relative squared error	0	%	
Total Number of Instances	200		

```
=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0	1	1	1	n
1	0	1	1	1	t

```
=== Confusion Matrix ===
```

```

  a   b   <-- classified as
100   0 |   a = n
  0 100 |   b = t

```

Pese a que los resultados arrojados pudieran parecer buenos, 100% de instancias bien clasificadas, no se debe tender a euforias, ya que si el sistema tiene sólo uno de profundidad, el vecino consultado será la propia instancia, siendo al final el valor decidido el valor de lla misma.

```
=== Evaluation on test split ===
=== Summary ===
```

Correctly Classified Instances	22	55	%
Incorrectly Classified Instances	18	45	%
Kappa statistic	0.0977		
Mean absolute error	0.45		
Root mean squared error	0.6708		
Relative absolute error	89.9445	%	
Root relative squared error	134.0711	%	
Total Number of Instances	40		

```
=== Detailed Accuracy By Class ===
```

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.526	0.429	0.526	0.526	0.526	n
0.571	0.474	0.571	0.571	0.571	t

```
=== Confusion Matrix ===
```

```

  a   b   <-- classified as
10   9 |   a = n
  9 12 |   b = t

```

El split demuestra lo antes mencionado al sólo clasificar el 55% bien de las instancias.

4.2.1.3.2. Clasificación pacientes

Finalmente se comprueba su resolución con el fichero test

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      146           48.6667 %
Incorrectly Classified Instances    154           51.3333 %
Kappa statistic                     0
Mean absolute error                 0.5133
Root mean squared error             0.7165
Relative absolute error             102.6667 %
Root relative squared error         143.2946 %
Total Number of Instances          300

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.487      0          1           0.487     0.655        n
  0          0.513      0           0         0           t

=== Confusion Matrix ===

  a    b    <-- classified as
146 154 |    a = n
  0   0  |    b = t
```

Como se puede observar este es el algoritmo que, de todos, más se aproxima a la solución ideal. De los 300 casos supuestos n, solo reconoce como tales 146, número muy próximo a los 150 ideales.

4.2.1.4. Algoritmo IBk

Se repite el mismo algoritmo anterior sólo que esta vez entrenando con K=3

4.2.1.4.1. Entrenamiento de la máquina

Los resultados del entrenamiento y del split se muestran a continuación:

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      160           80 %
Incorrectly Classified Instances    40           20 %
Kappa statistic                     0.6
Mean absolute error                 0.2757
Root mean squared error             0.3689
Relative absolute error             55.1495 %
Root relative squared error         73.7893 %
Total Number of Instances          200

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.8       0.2       0.8         0.8       0.8          n
  0.8       0.2       0.8         0.8       0.8          t
```

```

=== Confusion Matrix ===

  a  b  <-- classified as
 80 20 |  a = n
 20 80 |  b = t

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances      20          50      %
Incorrectly Classified Instances    20          50      %
Kappa statistic                    -0.0025
Mean absolute error                 0.4585
Root mean squared error             0.5447
Relative absolute error             91.6447 %
Root relative squared error        108.8574 %
Total Number of Instances          40

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.474      0.476      0.474      0.474      0.474        n
  0.524      0.526      0.524      0.524      0.524        t

=== Confusion Matrix ===

  a  b  <-- classified as
  9 10 |  a = n
 10 11 |  b = t

```

4.2.1.4.2. *Clasificación pacientes*

Finalmente, la clasificación es también mostrada.

```

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      129          43      %
Incorrectly Classified Instances    171          57      %
Kappa statistic                     0
Mean absolute error                 0.5321
Root mean squared error             0.6155
Relative absolute error             106.423 %
Root relative squared error        123.1059 %
Total Number of Instances          300

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.43       0          1          0.43      0.601        n
  0          0.57      0          0         0           t

=== Confusion Matrix ===

  a  b  <-- classified as
129 171 |  a = n
  0   0 |  b = t

```

Se puede comprobar que el porcentaje de acierto es más bajo que el obtenido en el caso de $K=1$. No siempre el consultar a tus vecinos te va a proporcionar información fidedigna sobre quien eres.

4.2.1.5. *Resultados finales*

En la siguiente tabla se muestra la clasificación de pacientes obtenida para los dos algoritmos que arrojaban mejores resultados. Para más información consultar el Anexo A con la clasificación de los 4 algoritmos anteriores.

J48				IB1			
Paciente	tipo	Paciente	tipo	Paciente	tipo	Paciente	tipo
1	T	16	N	1	N	16	T
2	N	17	N	2	N	17	N
3	T	18	N	3	N	18	T
4	T	19	N	4	T	19	N
5	N	20	T	5	N	20	T
6	T	21	N	6	T	21	T
7	N	22	T	7	N	22	T
8	N	23	N	8	T	23	N
9	N	24	T	9	T	24	N
10	N	25	T	10	T	25	N
11	N	26	T	11	N	26	T
12	T	27	N	12	N	27	T
13	T	28	T	13	T	28	T
14	T	29	T	14	N	29	T
15	N	30	T	15	N	30	N

Como se puede observar en la tabla, de los 30 pacientes, en 16 coinciden ambos algoritmos, lo que denota la falta de precisión de al menos uno de los dos algoritmos usados.

Por último comentar que aquellos casos dudosos donde existían los mismos indicios tanto para clasificarla como N o como para clasificarla como T, se han resuelto en beneficio de la clase perjudicada (es decir, aquella que menos veces hubiera sido clasificada).

4.2.2. *Test B: Clasificación en pacientes S y T*

En esta parte, el objetivo es conseguir clasificar las muestras contenidas en este test en los grupos de pacientes S y T.

Al cambiar la clase a detectar con respecto al caso anterior, será necesario cambiar el test de entrenamiento, por eso se usará st.arff en vez de nt.arff. Una vez entrenada el test se será clasificado.

Al igual que en el caso anterior, lo ideal sería obtener unos resultados de clasificación del 50% de los datos de cada tipo. A continuación se muestran los resultados obtenidos.

4.2.2.1. Algoritmo OneR

4.2.2.1.1. Entrenamiento de la máquina

Tras entrenar la máquina con el fichero adecuado se obtiene el siguiente resumen.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      139          69.5   %
Incorrectly Classified Instances    61           30.5   %
Kappa statistic                     0.39
Mean absolute error                 0.305
Root mean squared error            0.5523
Relative absolute error             61           %
Root relative squared error        110.4536 %
Total Number of Instances          200

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.73      0.34      0.682      0.73      0.705        s
  0.66      0.27      0.71       0.66      0.684        t

=== Confusion Matrix ===

  a  b  <-- classified as
73 27 |  a = s
34 66 |  b = t

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances      19          47.5   %
Incorrectly Classified Instances    21          52.5   %
Kappa statistic                    -0.0606
Mean absolute error                 0.525
Root mean squared error            0.7246
Relative absolute error            104.9352 %
Root relative squared error        144.8134 %
Total Number of Instances          40

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.368     0.429     0.438     0.368     0.4          s
  0.571     0.632     0.5       0.571     0.533        t

=== Confusion Matrix ===

  a  b  <-- classified as
 7 12 |  a = s
 9 12 |  b = t
```

4.2.2.1.2. Clasificación pacientes

A continuación, se procede a clasificar los pacientes del fichero test_b. Suponiendo a todos los pacientes t, estos son los resultados obtenidos en el procesamiento.

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      89          44.5   %
Incorrectly Classified Instances    111          55.5   %
Kappa statistic                     0
Mean absolute error                 0.555
Root mean squared error             0.745
Relative absolute error             111          %
Root relative squared error         148.9966   %
Total Number of Instances          200

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
    0         0.555         0           0           0           s
    0.445      0         1           0.445      0.616      t

=== Confusion Matrix ===

    a    b    <-- classified as
    0    0    |    a = s
  111   89    |    b = t
```

Como se puede observar el 55.5% de las muestras las clasifica como s, mientras que el 44.5% restante como t. Si se le compara con el 50 % ideal, no es un mal resultado, pero aún así y en vistas de lo obtenido en el entrenamiento cabe pensar que se habrá cometido algún error. Además, al solo evaluar un único atributo, es más probable que se cometan errores.

4.2.2.2. Algoritmo SMO

Se decidió recurrir a utilizar este algoritmo ante los malísimos resultados obtenidos al usar J48. Con J48 la máquina era incapaz de clasificar, clasificando directamente todos los pacientes como una de las dos clases posibles.

4.2.2.2.1. Entrenamiento de la máquina

Entrenando a la máquina con el algoritmo J48, se obtienen el siguiente resumen de resultados.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      108          54   %
Incorrectly Classified Instances     92          46   %
Kappa statistic                     0.08
Mean absolute error                 0.46
Root mean squared error             0.6782
```

```

Relative absolute error          92      %
Root relative squared error      135.6466 %
Total Number of Instances       200

```

```
=== Detailed Accuracy By Class ===
```

```

TP Rate   FP Rate   Precision   Recall   F-Measure   Class
  0.27     0.19     0.587     0.27     0.37        s
  0.81     0.73     0.526     0.81     0.638       t

```

```
=== Confusion Matrix ===
```

```

  a  b  <-- classified as
27 73 |  a = s
19 81 |  b = t

```

En este caso el 54% de las muestras están bien clasificadas. Al igual que antes, se procede a entrenar con el 80% y a probar con el 20% para ver si la máquina está tan bien entrenada.

```
=== Evaluation on test split ===
```

```
=== Summary ===
```

```

Correctly Classified Instances      16          40      %
Incorrectly Classified Instances    24          60      %
Kappa statistic                    -0.1736
Mean absolute error                 0.6
Root mean squared error             0.7746
Relative absolute error             119.926 %
Root relative squared error         154.812 %
Total Number of Instances          40

```

```
=== Detailed Accuracy By Class ===
```

```

TP Rate   FP Rate   Precision   Recall   F-Measure   Class
  0.632     0.81     0.414     0.632     0.5        s
  0.19     0.368     0.364     0.19     0.25       t

```

```
=== Confusion Matrix ===
```

```

  a  b  <-- classified as
12  7 |  a = s
17  4 |  b = t

```

4.2.2.2.2. Clasificación pacientes

Por último clasificamos el test obteniendo los resultados presentados a continuación

```
=== Evaluation on test set ===
```

```
=== Summary ===
```

```

Correctly Classified Instances      150          75      %
Incorrectly Classified Instances     50          25      %
Kappa statistic                     0
Mean absolute error                 0.25
Root mean squared error             0.5
Relative absolute error             50      %
Root relative squared error         100      %
Total Number of Instances          200

```

```

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
0           0.25         0            0          0            s
0.75        0           1            0.75       0.857        t

=== Confusion Matrix ===

  a    b    <-- classified as
  0    0    |    a = s
50 150    |    b = t

```

Se puede observar que el número de instancias t asignadas es mayor que el de instancias s.

4.2.2.3. Algoritmo IB5

4.2.2.3.1. Entrenamiento de la máquina

Al igual que en los apartados anteriores, antes de la clasificación del fichero test hay que entrenar la máquina. El algoritmo IBk fue seleccionado con K=5, es decir, profundidad 5. Los resultados tras el entrenamiento se muestran a continuación.

```

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      141           70.5   %
Incorrectly Classified Instances    59            29.5   %
Kappa statistic                     0.41
Mean absolute error                 0.3852
Root mean squared error             0.432
Relative absolute error             77.0459 %
Root relative squared error         86.3999 %
Total Number of Instances          200

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
0.65        0.24         0.73         0.65       0.688        s
0.76        0.35         0.685        0.76       0.72         t

=== Confusion Matrix ===

  a    b    <-- classified as
  65   35   |    a = s
  24   76   |    b = t

```

En este caso, el 70.5% de las muestras están bien clasificadas. Al igual que antes, se procede a entrenar con el 80% y a probar con el 20% para ver si en efecto la máquina está tan bien entrenada.

```

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances      22            55   %
Incorrectly Classified Instances    18            45   %

```

```

Kappa statistic                0.084
Mean absolute error            0.48
Root mean squared error        0.529
Relative absolute error         95.9507 %
Root relative squared error     105.7191 %
Total Number of Instances      40

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall   F-Measure   Class
0.368     0.286     0.538      0.368    0.438       s
0.714     0.632     0.556      0.714    0.625       t

=== Confusion Matrix ===

  a  b  <-- classified as
 7 12 |   a = s
 6 15 |   b = t

```

4.2.2.3.2. Clasificación pacientes

Finalmente se comprueba su resolución con el fichero test

```

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      110           55      %
Incorrectly Classified Instances    90           45      %
Kappa statistic                     0
Mean absolute error                 0.4561
Root mean squared error             0.5195
Relative absolute error              91.2176 %
Root relative squared error         103.8908 %
Total Number of Instances          200

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall   F-Measure   Class
0         0.45     0          0        0           s
0.55      0        1          0.55     0.71        t

=== Confusion Matrix ===

  a  b  <-- classified as
 0  0 |   a = s
 90 110 |   b = t

```

4.2.2.4. Resultados finales

En la siguiente tabla se muestra la clasificación de pacientes obtenida para los dos algoritmos que arrojaban mejores resultados. Para más información consultar el Anexo B con la clasificación de los 3 algoritmos anteriores.

OneR				IB5			
Paciente	tipo	Paciente	tipo	Paciente	tipo	Paciente	tipo
1	S	11	S	1	S	11	S
2	S	12	T	2	T	12	S
3	S	13	S	3	T	13	T
4	T	14	T	4	T	14	S

5	S	15	T	5	T	15	T
6	S	16	T	6	T	16	T
7	T	17	S	7	S	17	T
8	S	18	T	8	S	18	S
9	T	19	S	9	T	19	S
10	S	20	T	10	S	20	T

Como se puede observar en la tabla, de los 20 pacientes, en 10 coinciden ambos algoritmos, lo que denota una vez más, la falta de precisión de al menos uno de los dos algoritmos usados.

Por último comentar que con los casos dudosos se siguió el mismo criterio seguido en el test A

5. Conclusiones

En esta práctica se ha podido demostrar la gran utilidad que tiene la minería de datos, algo que se ha podido ver con un caso real y mediante la aplicación de diferentes algoritmos para el tratamiento de los datos desprendidos del problema.

En el caso de separar los pacientes N de los pacientes T, se pudo ver que los algoritmos que mejores resultados arrojaban fueron el J48 y el IB1. Como ya se dijo antes, el J48 se ajusta bastante bien al modelo que deseamos construir debido entre otras cosas a la poda que realiza una vez que el árbol ha sido creado.

En el otro caso, consistente en separar los pacientes S de los pacientes T, se pudo ver que los algoritmos que mejores resultados arrojaban fueron el IB5 y el OneR. Como ya se dijo antes, el J48 era incapaz de clasificar correctamente. Ya sea debido a la naturaleza de los datos, o a la simplicidad de los algoritmos utilizados los resultados arrojados en este segundo test son peores que los del primero.

Por último, comentar el gran número de algoritmos incluidos en WEKA, de los cuales, solo algunos eran compatibles con nuestros datos, muchos de ellos sólo trabajaban con valores numéricos como el ID3. También comentar que un estudio más exhaustivo de los registros, con selección y filtrado de datos, así como el estudio de la representación y segmentación de los datos de entrada, permitiría obtener mejores resultados. Es importante el algoritmo, pero puede que mucho más el preprocesamiento seguido; es más, se suelen hacer preprocesamientos pensados para un determinado algoritmo, lo cual es distinto de lo hecho en esta práctica ya que el mismo preprocesado se ha utilizado en los distintos algoritmos.

6. Bibliografía

Apuntes de clase

Practica Diagnóstico Cardiología 04/05 – A. Solano y N. Hernandez

Practica Diagnóstico Cardiología 04/05 – J.García y O. García

Practica Diagnóstico Cardiología 04/05 – M. Nuñez y A. Cantalapiedra

7. Anexo A: Tabla de resultados de la clasificación del test A

OneR			J48			IB1			IB3					
n	n	n	t	t	t	n	n	n	n	t	t			
n			n			n			t					
n			t			n			n					
t			t			t			t					
t			t			t			t					
n	n		t	n		t	n		n	n		n	t	t
t			n				t			t				
n			t				t			t				
n			n				n			t				
n			n				n			t				
n	n	n	n	n	n		n	n	n	n	t	x		

t			n			t			t		
n			t			n			n		
t			t			n			t		
n			n			t			t		
n	n		n	n		n	n		n	n	
t			n			t			n		
n			n			n			n		
t			n			t			t		
n			t			n			t		
t	n		t	n		t	t		n	t	
t			t			t			t		
n			n			n			t		
n			n			n			n		
n			t			n			n		
n	n		t	t		n	n		n	n	
t			t			n			t		
n			n			t			n		
t			n			t			t		
t			t			n			n		
t	t		t	t		t	t		t	t	
t			n			n			n		
t			n			t			t		
n			t			t			t		
n			t			t			t		
t	t		t	t		t	t		t	t	
n			n			n			n		
t			t			n			n		
t			n			n			n		
n			t			n			n		
n	t		n	t		t	n		t	t	
t			t			n			n		
t			n			t			t		
t			n			n			n		
t			t			n			n		
n	t		n	n		n	n		n	n	
t			t			n			n		
t			n			n			n		
t			t			n			n		
n			n			n			n		
n	t		n	n		t	t		n	t	
t			t			t			t		
t			n			t			t		
n			n			t			t		
n			t			t			t		
n	n		n	t		n	n		n	n	
n			t			n			n		
n			n			n			n		
n			t			n			n		
n			n			n			n		

t			n			n			n						
t			t			n									
n			n			n									
t	n	n	n	n	n	n	n	n	n	t	n				
n			n			t			t						
n			t			n			n						
t			t			t			t						
n			n			n			t						
n	n			n					n				t		
n				n					n				t		
t				t					t				n		
t				n					n				n		
n				n					n				n		
n	n	n	t	n	n	t	n	t	t	n	x				
n			n			t			t						
n			n			t			n						
n			n			t			t						
n			n			n			n						
n	n			n					t				t		
n				n					n				t		
t				n					t				n		
n				n					n				n		
n				n					n				n		
t	t	x	n	n	n	t	t	t	t	t	t				
t			n			t			t						
n			n			t			n						
t			n			t			t						
n			t			t			t						
t	n			t		t			t	t			t	t	
t				n					t				t		
n				n					t				n		
t				n					t				t		
n				n					t				t		
n	t	x	n	n	n	t	n	x	t	t	t				
t			n			t			t						
t			n			n			t						
n			n			n			n						
t			n			t			t						
t	n			n		t			t	t			t	t	
t				n					n				n		
t				n					t				t		
n				n					n				n		
n				t					t				t		
t	n	n	n	n	n	t	t	n	t	t	n				
n			t			n			n						
n			n			t			t						

n			n			n			n		
n			t			t			t		
n			n			n			n		
t			n			n			n		
n	n		n	n		n	n		n	n	
n			n			n			n		
n			t			n			n		
n											
n	n	n	t	n	x	n	n	x	t	t	t
n			t			n			n		
n			n			t			t		
n			n			n			n		
n			n			t			t		
n			t			n			n		
n	t		t	t		n	t		n	t	
t			n			t			t		
t			t			t			t		
t			t			n			n		
n		x	t		t	t		t	t		t
n			t			t			t		
n			t			t			t		
n			t			n			n		
n			t			t			t		
n			n	t		n	n		n	t	
t			t			n			n		
t			n			n			n		
t			t			t			t		
n	n	n	n	n	x	n	n	n	n	n	n
n			n			n			n		
n			n			t			n		
n			t			n			n		
n			n			n			n		
n			t			t			n		
n			n			n			n		
n			t			n			n		
n			n			n			n		
n	n	n	n	n	n	n	n	x	n	t	t
n			n			t			t		
n			t			t			t		
n			n			n			n		
t			t			n			t		
t			t			n			t		
t			n			n			n		
t			n			t			n		
n			n			n			n		
n			n			t			n		

n			n			n			n						
n	t	x	n	n	n	t	t	t	t	t	t				
t			n			t			t						
t			t			t			t						
t			n			n			n						
n			n			t			t						
n	n		n	n		n	n		n	t		t	n	t	t
n				n					t				t		
n				n					t				t		
t				n					t				t		
n				t					t				t		
n	n	n		t	n	n		n	x	x	n		n	t	
n				n				n			t				
n				n				n			n				
n				n				n			n				
n				t				t			t				
t	n		n	n	n		n	t	t		t	n	t		t
n				n				t				t			
n				n				t				t			
t				n				t				t			
n				t				t				t			
n	n	n		t	n	n		n	t	t		t	t	t	
n				t				t				t			
n				n				t				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	t	t		t	t	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	t	t		t	t	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n		n	n	n		n	n	t		t	n	t		t
n				t				t				t			
n				n				n				n			
n				t				t				t			
n				n				t				t			
n	n	n		n	n	n		n	t	t		n	t	t	
n				t				t				t			
n				n				n				n			
n				t				t				t			

n	t		n	t		t	t		t	t					
t			n			n									
t			n			n									
t			t			t									
t			t			t									
n	t	x	n	n	n	n	t	x	n	t	t				
n			n			n			n						
t			n			t			t						
t			t			t			t						
t			n			t			t						
n	n		n	n		t	n		t	t					
t			n			t			t						
n			t			n			n						
t			t			n			t						
n			n			n			n						
n	t	x	t	t	t	t	t	x	t	t	t				
t			t			t			t						
t			t			t			t						
n			t			n			n						
t			n			t			t						
n	n		t	t		n	n		t	t					
n			t			n			n						
n			t			n			n						
t			t			n			n						
t			n			t			n						
t	n	n	n	t	n	t	n	n	t	t	x				
n			t			n			t						
n			n			n			n						
n			t			t			t						
t			n			n			n						
n	n		n	n		n	t		n	t					
n			t			t			t						
n			n			n			n						
t			t			t			t						
n			n			n			n						
t	n	n	n	n	n	n	n	n	n	n	x				
n			n			n			n						
t			t			t			t						
n			n			n			n						
t			t			t			t						
n	n		t	n		n	t		t	t					
n			n			n			n						
n			t			t			t						
n			n			n			n						
n			t			t			t						
n	n	n	n	n	x	n	n	n	n	t	t				

n			t			t			t		
t			t			n			t		
t			n			n			t		
n			n			n			t		
n	n		t			n			n		
n			t			t			t		
n			t	t		t			n		
n			t			t			t		
n			n			t			n		
t			n			n			t		
n			t			n			n		
n	n		t			t			t		
n			n			t			t		
n			t	t		t			t		
n			t			t			t		
n			t			n			n		
n			t			n			t		
n			t			n			t		
n			t			n			t		
n	t		t			n			t		
n			t			n			t		
n			n			t			t		
n			t			n			t		
n			t			n			n		
n			t			n			t		
n			n			n			t		
n	n		n			n			t		
n			n			n			t		
t			t			n			t		
n			n			n			t		
t			n			n			n		
n			n			n			n		
n			n			n			t		
n	n		n			n			t		
n			n			n			t		
t			t			n			n		
n			n			n			n		
t			n			n			t		
n			n			n			t		
n			n			n			t		
n	n		n			n			n		
n			t			n			n		
n			n			n			n		
n			t			n			n		
n			n			n			n		
t			n			n			n		
n			n			n			t		
n	t		t			n			n		
t			t			t			t		
t			t			t			t		
n			n			t			n		
n			t			n			n		
t			t			t			t		
t			t			t			t		
n	t		n			n			n		
t			t			t			t		
t			t			t			t		
n			n			t			n		
n			t			n			n		
t			t			n			n		
t			t			t			t		

t			t			t			t						
t			n			n			n						
t			n			n			n						
n	n	x	n	n	t	t	n	n	t	t	x				
n			t			n			n						
t			t			n			t						
n			n			t			t						
t			n			n			n						
t	t			t		t			n	n			n	n	
t				t					n				t		
t				t					t				n		
n				n					n				n		
n				t					t				n		
n				t					n				n		
n				t					n				t		

8. Anexo B: Tabla de resultados de la clasificación del test B

OneR			IB5			SMO					
s	s	s	s	s	s	t	t	t			
s			s			t					
t			s			t					
s			t			t					
s			s			s					
s	s		s	s		t			t	t	
t			s	t							
s			t	s							
t			t	t							
s			t	t							
t	t	s	s	s	t	t	t	t			
s			s			s					

t			t			t		
s			t			t		
t			s			s		
s			t			t		
t			s			t		
s	s		t	t		s	t	
s			t			t		
s			t			s		
s			t			s		
t			s			s		
s	s		s	t		t	t	
s			t			t		
s		s	t		t	s		
t			t			t		
t			t			t		
s	s		s	t		t	t	
s			s			s		
t			t			t		
s			t			s		
t	t		t	t		s	t	
s			t			t		
t		t	t		t	t		x
s			t			s		
t			s	s		t	s	
t			s			s		
t			s			s		
s			s			s		
t			t			t		
s			s			t		
t			t			t		
s	s		t	t		t	t	
t		s	s		t	t		t
s			t			t		
t			t			t		
s			s			t		
s			t			t		
s	s		s	t		t	t	
s			t			t		
s			s			t		
s			t			t		
s			s			t		

t	t	t	s	s	t	t	t	t
t			s			t		
s			t			s		
s			s			t		
t			t			t		
t			t			s		
t			s			t		
s			t			t		
t			t			t		
s			t			s		
s	t	s	t	t	t	t	t	t
s			t			t		
t			s			t		
t			t			s		
t			s			t		
s			t			t		
s			t			t		
s			t			t		
s			s			s		
s			s			s		
t	t	t	s	s	x	s	t	t
s			t			t		
t			s			t		
s			s			t		
t			t			s		
t			s			t		
t			t			t		
s			t			t		
t			s			t		
s			t			t		
t	s	s	s	t	x	t	t	t
s			t			s		
s			t			t		
s			t			t		
t			s			s		
t			t			t		
s			s			t		
s			t			t		
s			s			t		
s			s			t		
s	s	x	t	s	x	t	t	t
s			t			t		
t			s			t		
s			s			t		
t			s			t		
t			t	t		s	t	

s			s			t		
s			s			t		
t			t			t		
t			t			t		