

**[Acta de evaluación]**

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

GRADO EN INGENIERÍA INFORMÁTICA

*Computación*

*Trabajo de Fin de Grado*

# PCTEnsembles: Integración de la consolidación parcial en varios ensembles de la plataforma Weka

Autor:

Jon Gámiz Aisa

2021/2022

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea

GRADO EN INGENIERÍA INFORMÁTICA

*Computación*

*Trabajo de Fin de Grado*

# PCTEnsembles: Integración de la consolidación parcial en varios ensembles de la plataforma Weka

Autor:

Jon Gámiz Aisa

2021/2022

# Agradecimientos

Gracias a las personas que han hecho posible que realice este proyecto.

Gracias a mis compañeros de clase por ayudarme y apoyarme en todo momento, a mis amigos por todo lo que me han ayudado para llegar hasta aquí y a mis padres, por apoyarme desde que soy pequeño en todos mis objetivos, en los buenos y malos momentos. Gracias a toda mi familia, mis abuelos, mis tíos, mis primos, gracias a todos. A toda la gente que me ha aguantado en este proceso.

Y sobre todo, dar las gracias a Txus Pérez como director del proyecto, por la paciencia que ha tenido conmigo y por no perder la esperanza. Todo el apoyo y la ayuda que me has dado y me has ofrecido no se paga con nada, sin ti no habría sido posible. Eskerrik asko!

# ÍNDICE

<b>RESUMEN</b>	<b>9</b>
<b>INTRODUCCIÓN</b>	<b>11</b>
<b>DOCUMENTO DE OBJETIVOS DEL PROYECTO</b>	<b>13</b>
<b>PUNTO DE PARTIDA</b>	<b>15</b>
<b>CONCEPTOS TEÓRICOS</b>	<b>17</b>
5.1 APRENDIZAJE AUTOMÁTICO	17
5.2 ÁRBOLES DE CLASIFICACIÓN	18
5.2.2 ALGORITMOS PRINCIPALES ÁRBOLES DE CLASIFICACIÓN	19
5.2.2.1 C45	19
5.2.2.2 J48	20
5.3 CLASIFICADORES MÚLTIPLES Ó ENSEMBLES [E INNER ENSEMBLES]	20
5.4 TÉCNICAS DE VALIDACIÓN	26
5.4.1 VALIDACIÓN CRUZADA	27
5.5 CRITERIOS DE BONDAD	27
<b>WEKA</b>	<b>32</b>
6.1 INTRODUCCIÓN	32
6.2 APLICACIONES Y MÓDULOS	33
6.3 DATOS	40
6.4 EXTENDIENDO WEKA	42
6.4.1 IMPLEMENTACIÓN	43
<b>PCT-ENSEMBLES</b>	<b>52</b>
<b>IMPLEMENTACIÓN DE PCTRSM Y PCTRF</b>	<b>54</b>
8.1 PCTRSM	54
8.2 PCTRF	68
<b>EXPERIMENTACIÓN</b>	<b>80</b>
9.1 EXPERIMENTACIÓN PARA ELEGIR LOS MEJORES RSM Y RF	80
9.1.1 CONCLUSIONES FASE 1	90
9.2 EXPERIMENTACIÓN PARA COMPARAR PCTRSM Y PCTRF CON SUS COMPETIDORES MÁS DIRECTOS	91
9.2.1 CONCLUSIONES FASE 2	110
<b>CONCLUSIONES Y LÍNEAS ABIERTAS</b>	<b>111</b>
10.1 LÍNEAS ABIERTAS	112
<b>BIBLIOGRAFÍA</b>	<b>113</b>
<b>APÉNDICE: TABLAS DE DATOS DE LA EXPERIMENTACIÓN</b>	<b>114</b>

# ÍNDICE DE FIGURAS

Figura 1: Árbol de decisión	17
Figura 2: Ejemplo de algoritmo Bagging	20
Figura 3: Ejemplo de Random Subspace Method	21
Figura 4: Ejemplo de Random Forest	22
Figura 5: Ejemplo de CTC - Construcción de un Árbol Consolidado [2]	23
Figura 6: Ejemplo de PCTBagging - Construcción de un árbol parcialmente consolidado	24
Figura 7: Ejemplo de la Validación Cruzada	25
Figura 8: Matriz de Confusión	26
Figura 9: Curva ROC	29
Figura 10: Interfaz gráfica de WEKA	32
Figura 11: Interfaz gráfica del modo Explorer - Preprocess	33
Figura 12: Interfaz gráfica del modo Explorer - Classify	34
Figura 13: Interfaz gráfica Experimenter - Setup	35
Figura 14: Interfaz gráfica del modo KnowledgeFlow	36
Figura 15: Interfaz gráfica del modo Simple CLI	38
Figura 16: Ejemplo de archivo .arff	40
Figura 17: Método globalInfo de J48	42
Figura 18: Método listOptions de J48	42
Figura 19: Método setOptions de J48	43
Figura 20: Método getOptions de J48	43
Figura 21: Método getCapabilities de J48	44
Figura 22: Método buildClassifier de J48	45
Figura 23: Método toString de J48	45
Figura 24: Método distributionForInstance de J48	46
Figura 25: Método classifyInstance de J48	46
Figura 26: Método main de J48	47
Figura 27: Interfaz gráfica para la elección de parámetros	48
Figura 28: PCTBconsolidationPercentTipText	49
Figura 29: getPCTBconsolidationPercent	49
Figura 30: setPCTBconsolidationPercent	49
Figura 31: Definición de la clase y sus atributos - PCTRSM	52
Figura 32: Método PCTRSMsubSpaceSizeTipText	53
Figura 33: Método getPCTRSMsubSpaceSize	54
Figura 34: Método setPCTRSMsubSpaceSize	54
Figura 35: Método PCTRSMsamplesGenerationTipText	55
Figura 36: Método getPCTRSMsamplesGeneration	55
Figura 37: Método setPCTRSMsamplesGeneration	55
Figura 38: Método listOptions - PCTRSM	56
Figura 39: Método setOptions - PCTRSM	57
Figura 40: Método getOptions - PCTRSM	57
Figura 41: Método globalInfo - PCTRSM	58

Figura 42: Primer segmento método buildClassifier - PCTRS	58
Figura 43: Segundo segmento método buildClassifier - PCTRS	59
Figura 44: Tercer segmento método buildClassifier - PCTRS	60
Figura 45: Cuarto segmento método buildClassifier - PCTRS	61
Figura 46: Quinto segmento método buildClassifier - PCTRS	62
Figura 47: Método numberOfAttributes - PCTRS	63
Figura 48: Método generateRSMSamples - PCTRS	64
Figura 49: Clase C45ToRSModelSelection - PCTRS	66
Figura 50: Definición de la clase y sus atributos - PCTRF	67
Figura 51: Método PCTRFnumFeaturesTipText	68
Figura 52: Método getPCTRFnumFeatures	68
Figura 53: Método setPCTRFnumFeatures	69
Figura 54: Método PCTRFsampleGenerationTipText	69
Figura 55: Método getPCTRFsamplesGeneration	69
Figura 56: Método setPCTRFsamplesGeneration	70
Figura 57: Método listOptions - PCTRF	71
Figura 58: Método setOptions - PCTRF	72
Figura 59: Método getOptions - PCTRF	73
Figura 60: Método globalInfo - PCTRF	73
Figura 61: Primer segmento método buildClassifier - PCTRF	74
Figura 62: Segundo segmento método buildClassifier - PCTRF	74
Figura 63: Tercer segmento método buildClassifier - PCTRF	75
Figura 64: Cuarto segmento método buildClassifier - PCTRF	75
Figura 65: Quinto segmento método buildClassifier - PCTRF	76
Figura 66: Primer segmento de C45ToRFModelSelection	77
Figura 67: Primer segmento de C45ToRFModelSelection	78
Figura 68: Gráfica Accuracy RF - Primera Fase	80
Figura 69: Gráfica Accuracy RSM y RF - Primera Fase	81
Figura 70: Gráfica Accuracy General - Primera Fase	81
Figura 71: Gráfica Balanced Accuracy RF - Primera Fase	82
Figura 72: Gráfica Balanced Accuracy RSM y RF - Primera Fase	83
Figura 73: Gráfica Balanced Accuracy General - Primera Fase	83
Figura 74: Gráfica AUC RF - Primera Fase	84
Figura 75: Gráfica AUC RSM y RF - Primera Fase	85
Figura 76: Gráfica AUC General - Primera Fase	86
Figura 77: Gráfica Kappa RF - Primera Fase	87
Figura 78: Gráfica Kappa RSM y RF - Primera Fase	88
Figura 79: Gráfica Kappa General - Primera Fase	88
Figura 80: Gráfica Accuracy PCTRS - Segunda Fase	91
Figura 81: Gráfica Accuracy PCTRF - Segunda Fase	92
Figura 82: Gráfica Accuracy General - Segunda Fase	92
Figura 83: Tabla de valores del Accuracy	93
Figura 84: Gráfica Balanced Accuracy PCTRS - Segunda Fase	94
Figura 85: Gráfica Balanced Accuracy PCTRF - Segunda Fase	94

Figura 86: Gráfica Balanced Accuracy General - Segunda Fase	95
Figura 87: Tabla de valores del Balanced Accuracy	95
Figura 88: Gráfica AUC PCTRSM - Segunda Fase	96
Figura 89: Gráfica AUC PCTRF - Segunda Fase	97
Figura 90: Gráfica AUC General - Segunda Fase	98
Figura 91: Gráfica Kappa PCTRSM - Segunda Fase	98
Figura 92: Gráfica Kappa PCTRF - Segunda Fase	99
Figura 93: Gráfica Kappa General - Segunda Fase	100
Figura 94: Gráfica Número de nodos PCTRSM - Segunda Fase	101
Figura 95: Gráfica Número de nodos PCTRF - Segunda Fase	102
Figura 96: Gráfica Número de nodos General - Segunda Fase	102
Figura 97: Tabla de valores del número de nodos del árbol	103
Figura 98: Gráfica Número de nodos hoja PCTRSM - Segunda Fase	103
Figura 99: Gráfica Número de nodos hoja PCTRF - Segunda Fase	104
Figura 100: Gráfica Número de nodos hoja General - Segunda Fase	105
Figura 101: Tabla de valores del número de nodos hoja del árbol	105
Figura 102: Gráfica Tiempo de construcción del clasificador PCTRSM - Segunda Fase	106
Figura 103: Gráfica Tiempo de construcción del clasificador PCTRF - Segunda Fase	107
Figura 104: Gráfica Tiempo de construcción del clasificador General - Segunda Fase	108
Figura 105: Tabla de valores del tiempo de construcción del clasificador	108



# 1. RESUMEN

El objetivo principal de este proyecto es integrar los algoritmos PCTRSM (Partially Consolidated Tree Random Subspace Method) y PCTRF (Partially Consolidated Tree Random Forest) diseñados por el grupo de investigación ALDAPA en la plataforma de código abierto WEKA.

El PCTRSM y el PCTRF son ambos una variación del algoritmo PCTBagging, en los que se incluyen los *ensembles* Random Subspace Method (RSM) y Random Forest (RF), respectivamente, en lugar del conocido clasificador múltiple o ensemble Bagging. El PCTBagging, es un híbrido entre el CTC (Árboles consolidados - Consolidated Tree Construction) y el Bagging (Árboles no consolidados), dependiendo del porcentaje de consolidación, 100% ó 0%, respectivamente. Si los árboles son (parcialmente) consolidados tendrán una estructura común y ésta proporcionará la explicación de la clasificación realizada.

Los ensembles RSM y RF introducen un factor de aleatoriedad en la construcción de los árboles que los componen, reduciendo (de diferente forma) el conjunto de variables a usar. Este mecanismo de aleatoriedad se puede integrar en el proceso de consolidación de los árboles y construir en WEKA los algoritmos propuestos en este trabajo: PCTRSM y PCTRF.

Se han realizado pruebas con 33 bases de datos donde se compararán los resultados obtenidos con el algoritmo CTC, el PCTBagging, PCTRSM, PCTRF y otros clasificadores con los que pueden competir. Todas estas pruebas y la experimentación se llevará a cabo en WEKA.

Palabras clave: algoritmo PCTRSM, PCTRF, WEKA, aprendizaje automático, árboles de clasificación, árboles consolidados, árboles parcialmente consolidados.



## 2. INTRODUCCIÓN

Los algoritmos PCTRSM y PCTRF tienen su origen en el algoritmo PCTBagging, que se sitúa en el paradigma de los árboles de clasificación, siendo este un híbrido entre el algoritmo CTC y el clasificador Bagging, en el que se construye un árbol consolidado parcial a partir de un conjunto de muestras, aplicando luego el Bagging y dejando que el resto de árboles se desarrollen independientemente.

El algoritmo CTC está diseñado por el grupo de investigación ALDAPA con el objetivo de construir un clasificador que tuviera en cuenta una distribución de clases distinta a la original, y además que el paradigma fuera explicativo, para poder proporcionar al usuario el motivo de la clasificación. Por ello el paradigma elegido fue el de los árboles de clasificación.

Un árbol consolidado es el resultado de un proceso en el que, al desarrollar un conjunto de árboles, todos toman la propuesta de división del nodo en curso más votada, que, dicho de otra manera, todos toman la propuesta de división por la que más árboles han optado. Por tanto, para poder realizar un árbol consolidado parcial es necesario un parámetro que indique en qué cantidad, o porcentaje del árbol se quiere consolidar. Este parámetro se llama Porcentaje de consolidación.

Los algoritmos PCTRSM y PCTRF funcionan de la misma manera que el PCTBagging, con el objetivo de añadir mayor aleatoriedad a su funcionamiento.

El algoritmo PCTRSM, al igual que el Random Subspace Method, genera un conjunto de atributos aleatorio y reducido a partir del conjunto de atributos original de los datos. Cada subconjunto de atributos se asignará a cada submuestra que se genera, siendo este nuevo conjunto de atributos el que se utilizará para la generación de los árboles de clasificación.

El algoritmo PCTRF, al igual que el Random Forest, genera un conjunto de atributos aleatorio y reducido a partir del conjunto de atributos original de los datos, de la misma manera que el PCTRSM, salvo que este nuevo subconjunto de atributos se generará para cada nodo de los árboles que se crean para cada submuestra.

WEKA es una plataforma de software libre muy popular la cual se centra en el Aprendizaje Automático y en la Minería de Datos, desarrollada en el lenguaje de programación JAVA. Para poder desarrollar el proyecto, en una primera parte se deberá analizar, comprender y estudiar la forma en la que la plataforma WEKA está implementada y posteriormente se procederá a integrar ambos algoritmos.



### **3. DOCUMENTO DE OBJETIVOS DEL PROYECTO**

Como se ha dicho anteriormente, el principal objetivo del proyecto es implementar los algoritmos PCTRSM y PCTRF integrándolos en la plataforma WEKA. La plataforma WEKA está perfectamente preparada para añadir nuevos algoritmos sin el requerimiento de modificar el código existente.

Los principales objetivos del proyecto son:

1. Conocer y familiarizarse con la plataforma WEKA, sabiendo cómo tienen estructuradas las diferentes secciones de la aplicación para posteriormente saber cómo integrar el algoritmo. Para ello documentarse de todas las clases de la aplicación y conocer el uso de cada una. De la misma manera, conocer y analizar las implementaciones en WEKA del CTC, el PCTBagging, el Random Subspace Method y el Random Forest.
2. Analizar y encontrar la forma de integrar los algoritmos en WEKA para posibilitar su uso a través de la interfaz gráfica de la aplicación e implementarlo.
3. Realizar una experimentación para determinar la validez de los algoritmos PCTRSM y PCTRF integrados en WEKA. Para ello, se utilizarán un conjunto de bases de datos y con ellas se harán una serie de pruebas. Se compararán los resultados con sus competidores más directos, clasificadores con los que compartan alguna similitud, para observar y analizar su comportamiento desde distintos puntos de vista.



## 4. PUNTO DE PARTIDA

El objetivo del proyecto es claro: diseñar, integrar y probar los algoritmos PCTRSM y PCTRF en WEKA.

Weka está implementado en el lenguaje de programación JAVA y tiene un sinfín de posibilidades en la rama de Aprendizaje automático que permiten a cualquier desarrollador probar sus propios algoritmos.

Centrándonos en los algoritmos de este proyecto, PCTRSM y PCTRF, las mayores fuentes de información obtenidas para realizar este proyecto son el artículo sobre el algoritmo PCTBagging [1] del grupo ALDAPA, en el cuál se basan PCTRSM y PCTRF, así como la tesis Doctoral de Jesús Ma Pérez de la Fuente [2] "Árboles consolidados: Construcción de un árbol de clasificación basado en múltiples submuestras sin renunciar a la explicación", en la que se explica el algoritmo CTC. Tanto el algoritmo PCTBagging como el CTC tienen una implementación en WEKA, siendo estas J48PartiallyConsolidated y J48Consolidated, respectivamente.

Y como el propio nombre de los algoritmos del proyecto indican, también se hace uso de los clasificadores Random Subspace Method [3] y Random Forest [4], diseñados por Ho y Leo Breiman, respectivamente.

Sabiendo todo esto como punto de partida para el proyecto, se diseñarán e implementarán los algoritmos PCTRSM y PCTRF y posteriormente se realizarán pruebas con un conjunto de bases de datos.





## 5. CONCEPTOS TEÓRICOS

En este capítulo, se explicarán los conceptos que son necesarios para entender y comprender mejor el contenido del proyecto.

### 5.1 APRENDIZAJE AUTOMÁTICO

El Aprendizaje Automático (*Machine Learning*) es una rama de la Inteligencia Artificial donde su objetivo es desarrollar técnicas que permitan a las máquinas *aprender*. Es una disciplina que, a través de algoritmos, dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones (análisis predictivo). Este aprendizaje permite a los computadores realizar tareas específicas de forma autónoma, es decir, sin necesidad de ser programados.

Esta rama del aprendizaje automático se puede aplicar y se aplica en muchas áreas fuera de la informática como, por ejemplo, son los diagnósticos médicos o las predicciones meteorológicas. Sirve para resolver una serie de problemas en los que se necesita predecir o clasificar. Para ello es importante obtener bien los datos antes mencionados, a esta disciplina de extraer los datos se le llama Minería de Datos (*Data Mining*).

Dentro del Aprendizaje Automático existen dos bloques, Aprendizaje Supervisado y No supervisado.

Este proyecto se centra en el Aprendizaje Supervisado, que cuentan con un conocimiento a priori que, dicho de otra forma, para clasificar o predecir un objeto dentro de una variable llamada clase, cuenta ya con unos casos ya clasificados.

En primer lugar, tenemos un conjunto de datos que le llamaremos Conjunto de Entrenamiento. Estos datos tienen una serie de atributos y también tienen la “resolución” del problema, es decir, tienen la variable clase definida.

Dentro de los algoritmos de Aprendizaje supervisado existen otros subgrupos, de los cuales nos centraremos en los Árboles de Clasificación.

## 5.2 ÁRBOLES DE CLASIFICACIÓN

Los Árboles de Clasificación son un método de Aprendizaje Automático que sirven para generar modelos de clasificación explicativos, es decir, el objetivo principal es el de crear un modelo, en este caso un árbol, el cual prediga el valor de la clase, en función de las diferentes variables de entrada. El árbol es la representación gráfica en el que las variables se van distribuyendo en función de unas condiciones que van cumpliendo las mismas.

En el proceso de construcción del árbol, todos los casos se sitúan en un único nodo llamado raíz, siendo este el árbol inicial. Se selecciona un atributo y se va dividiendo la población en función a la mejor condición obtenida por el algoritmo, creando así sus propios nodos posteriormente. De esta manera, dividiendo cada nodo con cada iteración se va creando un árbol. Cuando no se dan condiciones para hacer más divisiones, se para la ejecución en ese nodo y se convierte en nodo hoja. Los nodos hoja son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.

Una de las ventajas de los árboles de clasificación es su capacidad explicativa gracias al modelo que genera. Las redes neuronales, por ejemplo, no poseen ninguna capacidad explicativa, ya que realizan una clasificación o predicción sin explicar ni dar ninguna razón por la cual llega a una conclusión. En cambio los árboles, dado un caso a clasificar, el resultado o la decisión del árbol es una de las hojas del mismo, y el camino desde la raíz es la explicación de la clasificación. Como se puede observar en la figura 1, partiendo desde el nodo raíz, podemos ver qué decisiones toma el clasificador a la hora de devolver su predicción en función del valor de las variables.

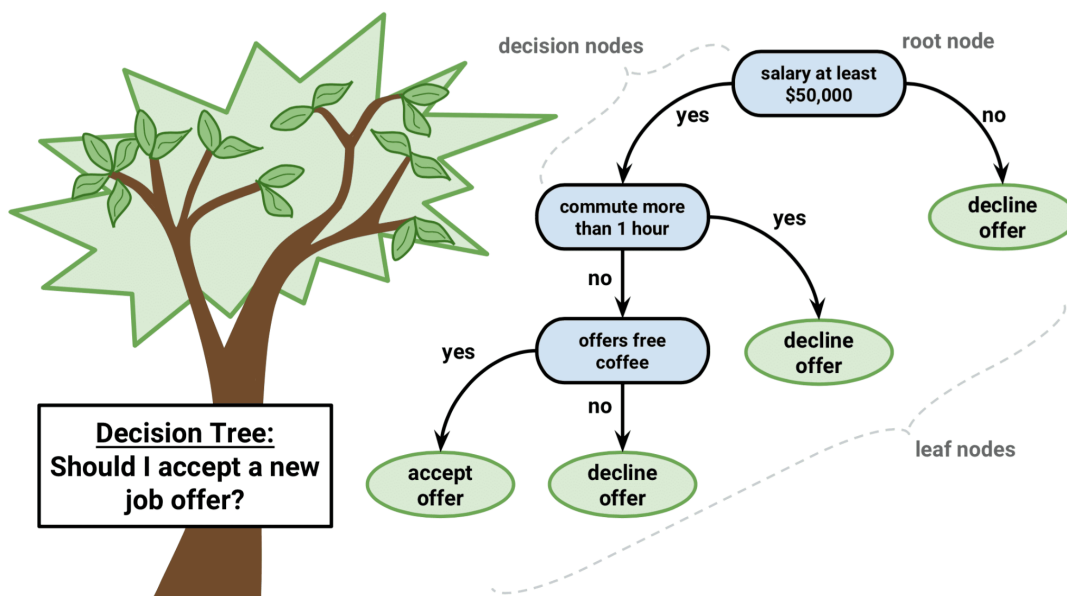


Figura 1: Árbol de decisión

[Fuente: <https://www.datacamp.com/tutorial/decision-trees-R>]

## 5.2.2 ALGORITMOS PRINCIPALES ÁRBOLES DE CLASIFICACIÓN

Dentro de la rama de los árboles de clasificación, los tres algoritmos más extendidos son el *C4.5*, *CART* y *CHAID*. Para este proyecto, nos vamos a centrar en todo momento en el *C4.5*, siendo este algoritmo la base que se va utilizar para cualquier generación de árboles.

### 5.2.2.1 C45

El algoritmo *C4.5* es uno de los algoritmos más utilizados en el ámbito de los árboles de clasificación, elaborado por Ross Quinlan [6]. Este algoritmo es una extensión de otro algoritmo, el algoritmo *ID3*. El algoritmo *C4.5* construye árboles de clasificación a partir de un conjunto de entrenamiento, de la misma manera que el algoritmo *ID3*, usando el concepto de entropía y ganancia de la información.

La ganancia de información es la propiedad estadística que mide qué tan bien un atributo dado separa los ejemplos de entrenamiento de acuerdo con su clasificación objetivo.

El concepto de la entropía, inventado por Claude E. Shannon, mide la impureza del conjunto de entrada. En física y matemáticas, la entropía se conoce como aleatoriedad o impureza en el sistema. En teoría de la información, se refiere a la impureza en un grupo de ejemplos. La ganancia de información es una medida de disminución de la entropía.

Los datos de entrenamiento son un conjunto  $S = s_1, s_2, \dots$  de los casos o datos que ya han sido clasificadas. Cada caso  $S_i = x_1, x_2, \dots$  es un vector, donde  $x_1, x_2, \dots$  representan los atributos o características de la muestra. Los datos de entrenamiento se ven aumentados con un vector  $C = c_1, c_2, \dots$  donde  $c_1, c_2, \dots$  representan la clase a que pertenece.

Para cada nodo del árbol, el algoritmo *C4.5* elige el atributo que más eficazmente lo divide. Su criterio es el de ganancia de información que dependiendo de los resultados tomará la elección de un atributo para dividir los datos. El atributo con la mayor ganancia será el que se elija para tomar la decisión. El algoritmo *C4.5* luego seguirá recursivamente en los siguientes nodos hijos.

### 5.2.2.2 J48

El algoritmo *J48* simplemente es una implementación libre del *C4.5* la cual está integrada en la plataforma *WEKA*. El creador de dicha implementación del algoritmo *J48* es Eibe Frank, y es el código sobre el que se basan los algoritmos *CTC* (*J48Consolidated*), *PCTBagging* (*J48PartiallyConsolidated*), *PCTRSM*, *PCTRF*...

## 5.3 CLASIFICADORES MÚLTIPLES Ó ENSEMBLES [E INNER ENSEMBLES]

Una de las técnicas de aprendizaje automático más poderosas es el aprendizaje por conjuntos. El *Ensemble Learning* (aprendizaje por conjuntos), es el proceso de entrenar múltiples modelos de aprendizaje automático y combinar sus resultados. Los diferentes modelos se utilizan como base para crear un modelo predictivo óptimo. La combinación de un conjunto diverso de modelos de aprendizaje automático individuales puede mejorar la estabilidad del modelo general y generar predicciones más precisas y suelen ser más fiables que los modelos individuales.

Los algoritmos desarrollados en este proyecto, PCTRSM y PCTRF, están basados en los algoritmos que se explican a continuación, todos ellos pertenecientes a la familia de los Ensembles, la construcción de varios clasificadores, es decir, clasificadores múltiples.

### Bagging

El Bagging, cuyo nombre proviene de Bootstrap aggregating (Agregación del Bootstrap)[5], es un meta-algoritmo de clasificación diseñado por Leo Breiman en 1994, para mejorar la estabilidad y precisión de algoritmos de aprendizaje automático usados en clasificación estadística y regresión, reduciendo la varianza y ayudando a evitar el sobreajuste.

Como el nombre Agregación del Bootstrap indica, se utiliza remuestreo bootstrap para añadir más muestras, realizando un remuestreo a partir del dataset original, siendo estas nuevas muestras uniformes y del mismo tamaño del original, con repeticiones y reemplazos de los datos. Una vez hecho esto, a partir de cada muestra se construye un modelo independiente con un clasificador simple y a partir de ahí se ensambla el modelo Bagging, clasificando con todos los modelos y asignándose la clase más votada, formando así un clasificador robusto. Siempre se utiliza el mismo algoritmo de aprendizaje con cada muestra. Bagging fue propuesto con los árboles de decisión C4.5, que ha sido el algoritmo más usado, pero también se han usado otros como RNAs, SVMs...

En la figura 2 se puede observar un esquema de este meta algoritmo.

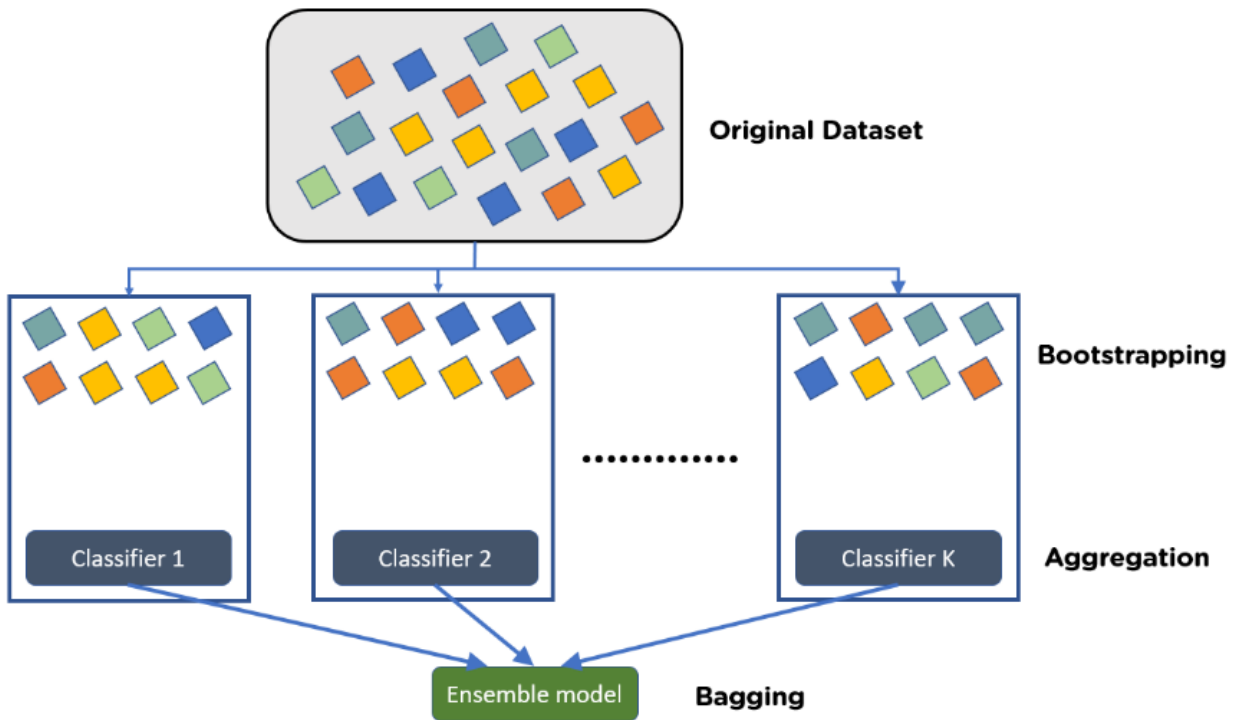


Figura 2: Ejemplo de algoritmo Bagging

[Fuente: [https://en.wikipedia.org/wiki/Bootstrap\\_aggregating](https://en.wikipedia.org/wiki/Bootstrap_aggregating)]

## Random Subspace Method

El Random Subspace Method (RSM) es una técnica de aprendizaje por conjuntos que fue propuesta por Ho. El principio es aumentar la diversidad entre los casos de la base de datos restringiendo los clasificadores para que trabajen en diferentes subespacios aleatorios del espacio completo de atributos.

El algoritmo no utiliza un remuestreo para generar varias muestras, utiliza la base de datos original sin ninguna modificación el número de veces deseado. Como se puede observar en la figura 3, cada clasificador aprende con una de estas muestras a la que se le aplica un subconjunto de atributos de tamaño  $n$ , es a lo que se llama subespacio, que es elegido uniformemente al azar con el uso de un mismo algoritmo para todos ellos a partir del conjunto completo de tamaño  $N$ . Al igual que Bagging, clasificará en función a la clase más votada por todos los clasificadores. Fue propuesto con el algoritmo C4.5, pero puede usarse con cualquier algoritmo base.

El RSM puede beneficiarse del uso de subespacios aleatorios, tanto para la construcción como la agregación de los clasificadores. Cuando el número de casos de entrenamiento es relativamente pequeño en comparación con el total de atributos de los datos, mediante la construcción de clasificadores en subespacios generados de forma aleatoria, se puede resolver el problema del tamaño de una base de datos pequeña. El tamaño del subespacio es más pequeño que el espacio de atributos original, mientras que la cantidad de objetos de entrenamiento permanece lo mismo. Por lo tanto, el tamaño relativo de la muestra de entrenamiento aumenta.

Cuando los datos tienen muchas características redundantes, se pueden obtener mejores clasificadores en subespacios aleatorios que en el espacio de características original. La decisión conjunta de tales clasificadores puede ser superior a un solo clasificador construido en el conjunto de entrenamiento original con el espacio de atributos completo.

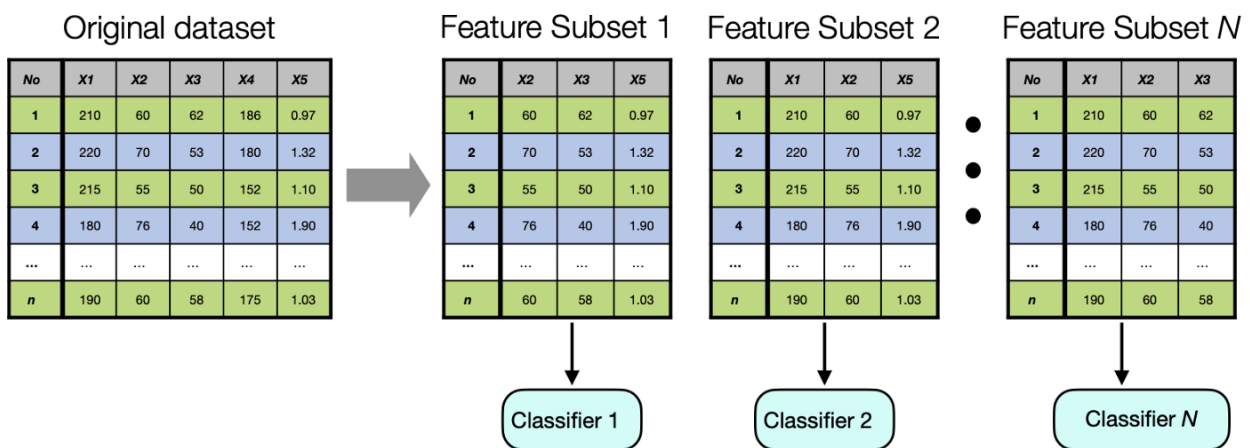


Figura 3: Ejemplo de Random Subspace Method

[Fuente: <https://towardsdatascience.com/ensembles-in-machine-learning-9128215629d1>]

## Random Forest

Random Forest, también conocido en castellano como “Bosques Aleatorios”, es una combinación de árboles predictores tal que los nodos de cada árbol dependen de unos número de atributos menor al total de los atributos generados aleatoriamente. Es una modificación sustancial de Bagging que construye una larga colección de árboles no correlacionados y luego los promedia.

El algoritmo para inducir un Random Forest fue desarrollado por Leo Breiman y Adele Cutler y Random Forest es su marca de fábrica. El método combina la idea de bagging de Breiman y la selección aleatoria de atributos, introducida independientemente por Ho, Amit y Geman, para construir una colección de árboles de decisión con variación controlada.

La selección de un subconjunto aleatorio de atributos es un ejemplo del método *Random Subspace Method*, como se ha comentado en el apartado anterior.

En muchos problemas el rendimiento del algoritmo Random Forest es muy similar a la del Boosting, un meta-algoritmo bastante conocido y usado, pero este es más simple de entrenar y ajustar. Como consecuencia, el Random Forest es popular y ampliamente utilizado.

Cada árbol es construido usando el siguiente algoritmo, como se puede ver en la figura 4:

1.  $K$  es el número de variables en el clasificador.
2. Sea  $k$  el número de variables de entrada a ser usado para determinar la decisión en un nodo dado;  $k$  debe ser menor que  $K$ .
3. Para cada nodo del árbol, elegir aleatoriamente  $k$  variables en las cuales basar la decisión. Calcular la mejor partición del conjunto de entrenamiento a partir de las  $k$  variables.

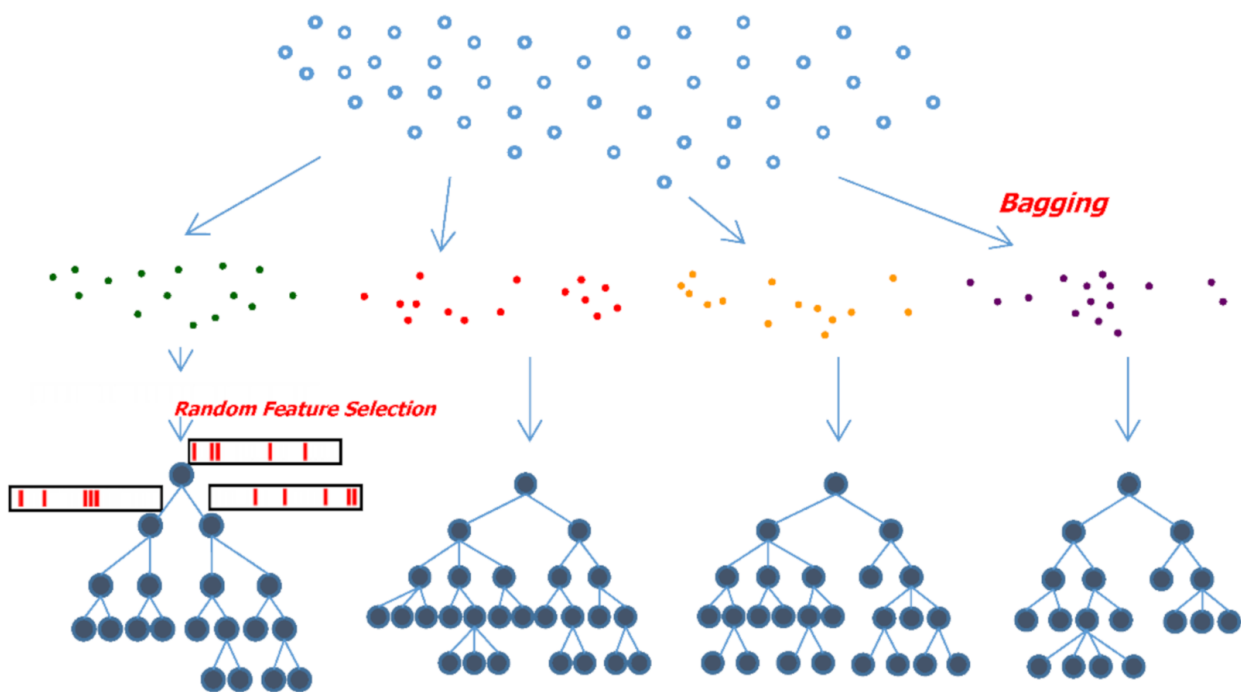


Figura 4: Ejemplo de Random Forest

[Fuente: [https://www.mdpi.com/2227-7102/11/3/92?type=check\\_update&version=1](https://www.mdpi.com/2227-7102/11/3/92?type=check_update&version=1)]

# CTC

El algoritmo CTC (Consolidated Trees' Construction) primero genera un conjunto de muestras y posteriormente se va construyendo un árbol C4.5 asociado a cada muestra. Mediante consenso, se va decidiendo qué variable es la más apropiada por la que dividir cada nodo, siendo esta una variable consolidada. Tras esta decisión, todas las muestras se dividen obligatoriamente por esa variable consolidada, construyendo así nodo por nodo el Árbol Consolidado, como se puede observar en la figura 5.

El algoritmo fue diseñado por el grupo de investigación ALDAPA basándose en el algoritmo C4.5. El propio grupo de investigación implementó el algoritmo en una plataforma llamada HARITZA, y Fernando Lozano en su proyecto de fin de grado implementó una aproximación en la plataforma Weka, llamada J48Consolidated.

Con el CTC se añadió una estrategia de remuestreo para algoritmos de clasificación que utilizan múltiples submuestras, llamada coverage. Esta estrategia se basa en la distribución de clases de la muestra de entrenamiento para garantizar una representación mínima de todas las clases al volver a muestrear. Con este valor, no hay que indicar el número de muestras que quieren generar, es mediante esta estrategia como se calcula de manera interna.

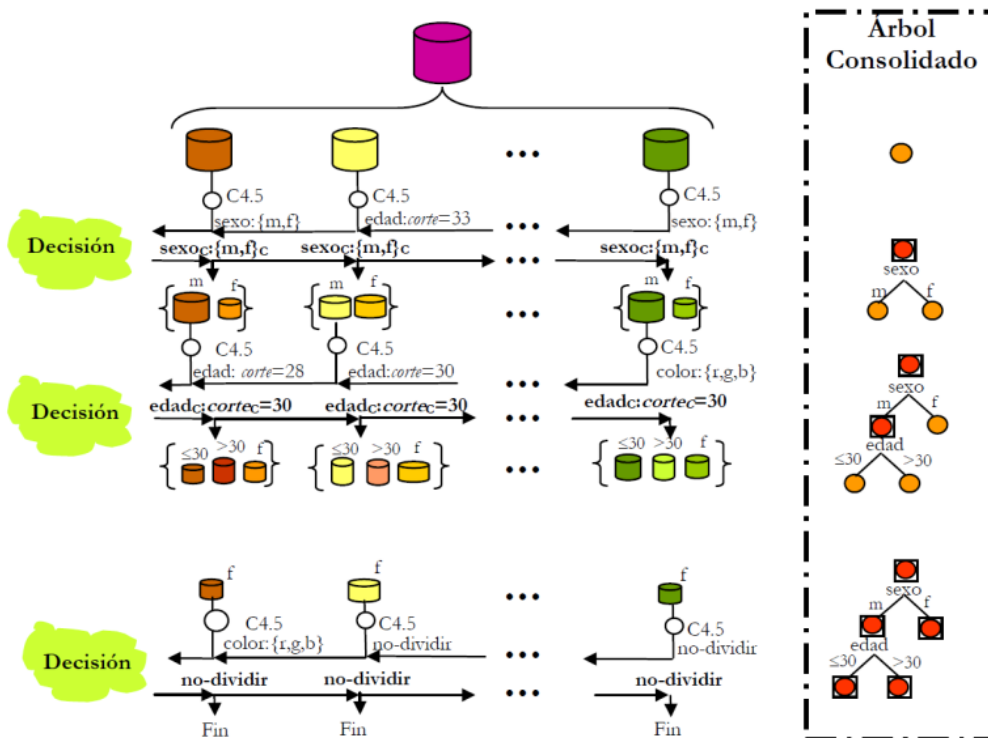


Figura 5: Ejemplo de CTC - Construcción de un Árbol Consolidado [2]



## PCTBagging

El algoritmo PCTBagging fue diseñado por el grupo de Investigación de ALDAPA, e implementado en WEKA como proyecto de fin de grado de Ander Otsoa con el nombre de J48PartiallyConsolidated.

El PCTBagging, es un híbrido entre el CTC (Árboles consolidados - Consolidated Tree Construction) y el Bagging (Árboles no consolidados), dependiendo del porcentaje de consolidación, 100% ó 0%, respectivamente. Si los árboles son (parcialmente) consolidados tendrán una estructura común y ésta proporcionará la explicación de la clasificación realizada. Lo que se busca es un compromiso entre la capacidad clasificatoria y explicativa.

Este algoritmo aplica el algoritmo CTC obteniendo el árbol consolidado. Después se poda dicho árbol y los subárboles generados de cada submuestra dependiendo del Porcentaje de Consolidación. Después a cada subárbol se le deja desarrollarse independientemente (sin consolidación) constituyendo con todos los árboles un Bagging.

En la figura 6 se puede ver un ejemplo de árbol parcialmente consolidado, a partir del cual se generan varios árboles individuales sin relación ni consolidación entre ellos.

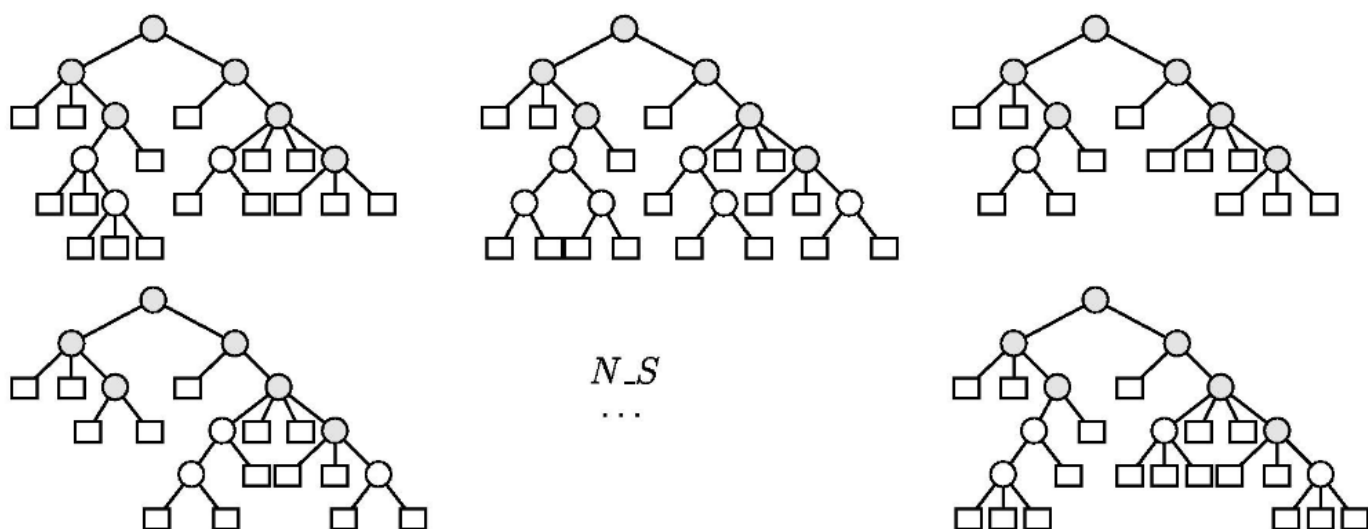


Figura 6: Ejemplo de PCTBagging - Construcción de un árbol parcialmente consolidado

[Fuente: <https://www.sciencedirect.com/science/article/pii/S0020025521011208>]

## 5.4 TÉCNICAS DE VALIDACIÓN

La validación de métodos es el proceso utilizado mediante las técnicas de validación para confirmar que el procedimiento analítico empleado para una prueba específica es adecuado para el uso previsto. Los resultados de la validación del método pueden usarse para juzgar la calidad, confiabilidad y consistencia de los resultados, es una parte integral de cualquier buena práctica analítica.

Las técnicas de validación en el aprendizaje automático se utilizan para obtener, por ejemplo, la tasa de error del modelo generado, que se puede considerar cercana a la verdadera tasa de error de la población. Si el volumen de datos es lo suficientemente grande como para ser representativo de la población, es posible que no necesite las técnicas de validación. Sin embargo, en escenarios del mundo real, trabajamos con muestras de datos que pueden no ser un verdadero representante de la población. Aquí es donde las técnicas de validación entran en escena.

### 5.4.1 VALIDACIÓN CRUZADA

La validación cruzada o *cross-validation* es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de los criterios de bondad sobre diferentes particiones, es una técnica muy utilizada para validar modelos. Hay varias formas de realizar las particiones de datos de entrenamiento y prueba, ya sea dejando como dato de prueba un único valor, de manera aleatoria o con “ $k$ ” iteraciones, como se puede ver en la figura 7, siendo esta última la utilizada para este proyecto.

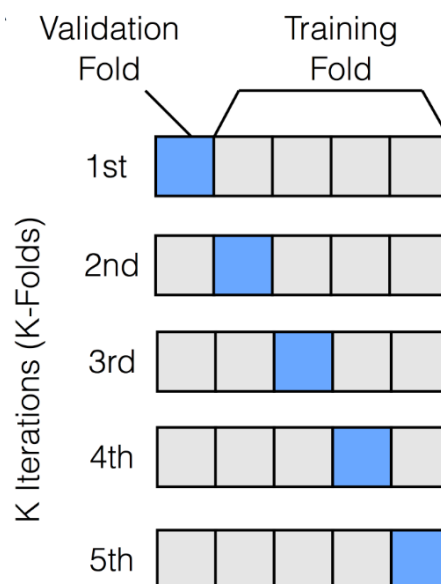


Figura 7: Ejemplo de la Validación Cruzada

[Fuente: <https://androidkt.com/pytorch-k-fold-cross-validation-using-dataloader-and-sklearn/>]

## 5.5 CRITERIOS DE BONDAD

Los criterios de bondad son los criterios o valores que se utilizarán para medir el rendimiento de nuestros modelos aplicados en unas bases de datos.

En esta sección se van a introducir brevemente los criterios utilizados en la experimentación para evaluar el rendimiento del PCTRSM y del PCTRF junto a otros algoritmos con los que compiten.

Estos criterios son utilizados en problemas clásicos, en los que se centran los algoritmos PCTRSM y PCTRF, es decir, problemas con 2 clases posibles. Se van a dividir en tres tipos según lo que evalúan: capacidad de clasificación, capacidad explicativa y coste computacional.

### Medidas de Capacidad de Clasificación

Como el nombre indica, estas son las medidas utilizadas para saber cómo de bien clasifica el clasificador desde distintas perspectivas. Muchos de estos criterios utilizan elementos de la tabla mostrada en la figura 8, llamada Matriz de confusión:

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Figura 8: Matriz de Confusión

[Fuente: <https://www.quora.com/What-is-a-confusion-matrix-in-data-science>]

Las filas hacen referencia a los valores que ha predicho el algoritmo utilizado, y las columnas son el valor de la clase real para ese caso. La tabla se divide en 4 opciones posibles:

- **TP o True Positive:** Muestra el número de casos cuya clase real es positiva, y nuestro clasificador ha acertado en su predicción, ha clasificado como positiva.
- **FP o False Positive:** Muestra el número de casos cuya clase real es negativa, y nuestro clasificador ha fallado en su predicción, ha clasificado como positiva.
- **FN o False Negative:** Muestra el número de casos cuya clase real es positiva, y nuestro clasificador ha fallado en su predicción, ha clasificado como negativa.
- **TN o True Negative:** Muestra el número de casos cuya clase real es negativa, y nuestro clasificador ha acertado en su predicción, ha clasificado como negativa.

A continuación se mostrarán, haciendo uso de los valores de la Matriz de confusión, los 4 criterios utilizados para evaluar la capacidad de clasificación de los clasificadores:

## Accuracy

La Tasa de Acierto o *Accuracy* se refiere a lo cerca que está el resultado de una medición del valor verdadero. Se representa como la proporción de resultados verdaderos (tanto verdaderos positivos (TP) como verdaderos negativos (TN)) con respecto al número total de casos examinados (verdaderos positivos, falsos positivos, verdaderos negativos, falsos negativos)

En forma práctica, la Tasa de Acierto es la cantidad de predicciones, tanto positivas como negativas, que fueron correctas.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Las predicciones correctas, las muestras True Positive y True Negative, se dividen entre el total de los casos, para obtener el porcentaje de acierto.

Aunque como primer contacto puede estar bien, este criterio puede no ser suficiente en caso de que las clases estén desbalanceadas, y puede darnos una idea equivocada sobre cómo de bien funciona nuestro algoritmo.

Una muestra desbalanceada es un conjunto de datos en los cuales la clase está muy desproporcionada, muchos ejemplos para una clase pero muy pocos para la otra. Un clasificador que siempre clasifique cualquier caso como la clase mayoritaria tendrá muy buenos resultados con una muestra de este tipo, pero realmente no resuelve el problema.

## Balanced Accuracy

Esta es la medida más adecuada para evaluar un algoritmo en bases de datos desbalanceadas. Al contrario que el Accuracy, el Balanced Accuracy califica la tasa de acierto calculando la media de acierto de los positivos y negativos individualmente, o dicho de otra manera, de forma balanceada, para evitar los problemas que una gran diferencia de casos positivos y casos negativos supone.

A la media de acierto de los positivos se llama Sensibilidad, siendo la fórmula:

$$\text{Sensibilidad} = TP / (TP + FN)$$

La media de los negativos, en cambio se llama Especificidad y sigue el mismo proceso:

$$\text{Especificidad} = TN / (TN + FP)$$

Luego el Balanced Accuracy es la media aritmética de ambas medidas:

$$\text{Balanced Accuracy} = (\text{Sensibilidad} + \text{Especificidad}) / 2$$

## Kappa

Esta es la medida que indica cuánto mejora el clasificador respecto a uno que clasifica aleatoriamente acorde a la frecuencia de la clase. La ecuación base de la medida es esta:

$$\text{Kappa} = (\text{Pr}(a) - \text{Pr}(e)) / (1 - \text{Pr}(e))$$

donde  $\text{Pr}(a)$  es el acuerdo observado relativo entre los observadores, y  $\text{Pr}(e)$  es la probabilidad hipotética de acuerdo por azar, utilizando los casos para calcular las probabilidades de que cada observador clasifique aleatoriamente. Esta medida solo mide el acuerdo entre dos observadores.

Ahora poniendo esto desde la perspectiva de una Matriz de confusión, se considera  $\text{Pr}(a)$  como el Accuracy ya que representa la concordancia entre las predicciones y la clase real.

El  $\text{Pr}(e)$  depende de las elecciones de cada observador, para calcular la probabilidad de que las predicciones sean positivas es:

$$\text{P(Positiva Observador)} = (TP + FP) / (TP + FP + FN + TN)$$

Y la probabilidad de que la clase real sea positiva es:

$$P(\text{Positiva Real}) = (TP + FN) / (TP + FP + FN + TN)$$

Por tanto, si se quiere obtener la probabilidad de que ambos sean positivos al azar es la multiplicación de ambas.

$$P(\text{Positiva Observador y Real}) = P(\text{Positiva Observador}) * P(\text{Positiva Real})$$

Posteriormente, para realizar el cálculo de la probabilidad de que ambos sean negativos, se realiza el procedimiento análogo correspondiente. Por tanto, siendo el valor de  $Pr(e)$  la probabilidad de que los valores elegidos aleatoriamente por ambos observadores sean positivos, por la probabilidad de que los valores elegidos aleatoriamente por ambos observadores sean negativos.

$$P(e) = P(\text{Positiva Observador y Real}) * P(\text{Negativa Observador y Real})$$

Una vez conocido cómo calcular tanto  $Pr(a)$  cómo  $Pr(e)$  con los valores de la Matriz de confusión, la ecuación final para calcular la medida kappa sería la siguiente:

$$Kappa = (2 * (TP * TN - FN * FP)) / ((TP + FP) * (FP + TN) + (TP + FN) * (FN + TN))$$

### AUC (Area under ROC curve)

AUC ponderada, también abreviado como AUC, es una mejor manera de medir los clasificadores de aprendizaje de datos desequilibrados. Se centra en la tasa de acierto de la clase minoritaria al variar los pesos con los valores del TPR (True Positive Rate) bajo la curva ROC.

La curva ROC es una gráfica, como se puede observar en la figura 9, que compara el TPR (True Positive Rate) y el FPR (False Positive Rate) en diferentes umbrales de la clasificación.

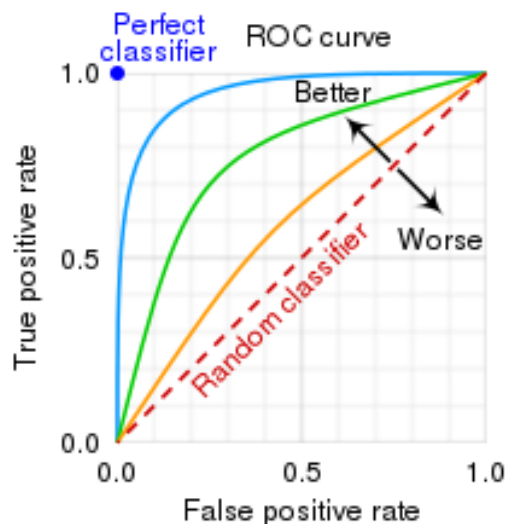


Figura 9: Curva ROC

[Fuente: [https://commons.wikimedia.org/wiki/File:Roc\\_curve.svg](https://commons.wikimedia.org/wiki/File:Roc_curve.svg)]

## **Medidas de Capacidad Explicativa**

Los siguientes criterios miden la complejidad del proceso de clasificación a la hora de poder dar explicaciones al usuario y la capacidad de visualizar los resultados y entender las razones que han llevado al clasificador a clasificar un caso de una manera u otra. Para nuestros clasificadores, estas medidas se centran en cómo es el árbol resultado respecto a su complejidad.

### **Número de nodos**

Cada nodo tiene una variable por la cual se divide la población, a no ser que sea un nodo hoja, siendo el número de estos nodos internos una de las principales medidas para conocer la complejidad del árbol, ya que, siendo el objetivo poder dar al usuario la posibilidad de entender el razonamiento de la clasificación de los casos, cuanto mayor cantidad de nodos hay, se crean más ramificaciones o divisiones y el árbol se vuelve menos explicativo y complicado de entender o descifrar.

### **Número de nodos hoja**

Los nodos hoja son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura. Ya que el objetivo es dar la explicación más sencilla posible para facilitar la comprensión de la clasificación, cuanto menor sea el número de nodos hoja, más explicativo será de entender.

## **Medidas de coste computacional**

Como su nombre indica, estas medidas permiten saber cuánta carga computacional necesitan los algoritmos sobre los ordenadores en los que se ejecutan, así como el tiempo que pueden tardar en distintos momentos del proceso.

### **Tiempo de construcción del clasificador**

Llamado UserCPU Time Training en WEKA, mide el tiempo que tarda en construir el clasificador en la CPU del ordenador del usuario en el que se ha ejecutado.





## 6. WEKA

En este capítulo se va a hablar de Weka, de sus características y funciones principales. Se intentará también explicar qué soluciones y ventajas ofrece en el ámbito del Aprendizaje Automático.

### 6.1 INTRODUCCIÓN

Weka (Waikato Environment for Knowledge Analysis - Entorno para Análisis del Conocimiento de la Universidad de Waikato) es una plataforma de software libre la cual se centra en el Aprendizaje Automático y en la Minería de Datos.

Weka es un software el cual contiene una colección de herramientas propias que permiten visualizar, ejecutar y añadir algoritmos para el análisis de datos y de modelo predictivo. El proyecto Weka comenzó en el año 1993 en la Universidad de Waikato en Nueva Zelanda. Este primer proyecto comenzó a programarse en TCL y C, aunque en 1997 se replantearon la opción de programarlo en Java y reescribieron todo el código que habían generado para volverlo al lenguaje de programación Java. Hoy en día sigue dándose soporte a Weka en Java.

El uso de Weka a nivel de usuario no es muy complicado, pero se requiere un conocimiento básico de Aprendizaje Automático y Minería de Datos. Añadido a esto, también incluye una interfaz gráfica (GUI - Graphical User Interface) de usuario para acceder y configurar las diferentes herramientas integradas de una manera intuitiva y fácil de usar, con el objetivo de facilitar el uso de Weka a un mayor número de usuarios.

Una de las grandes ventajas de WEKA, es que al ser GNU o, dicho de otra forma, código abierto, cualquier usuario puede modificar o añadir código a su gusto, pudiendo así adaptarlo a las necesidades particulares de cada usuario. Es gracias a esto que este proyecto se haya podido realizar, ya que se ha modificado el código existente para añadir dos nuevos algoritmos de aprendizaje supervisado: PCTRSM y PCTRF.

Weka es un software muy utilizado, es por eso que está en constante actualización, cada año publican varias versiones que mejoran el funcionamiento de la aplicación, actualmente está por la versión 3.8.6.

A continuación se describirán brevemente las diferentes opciones que Weka da a los usuarios.

## 6.2 APLICACIONES Y MÓDULOS

Como vemos en la figura 10, al iniciar Weka, la interfaz gráfica nos brinda la posibilidad de elegir entre cinco aplicaciones: *Explorer*, *Experimenter*, *Knowledgeflow*, *Workbench* y *Simple CLI*. Además nos encontramos en la barra de menú superior otras opciones que son secundarias en el programa como pueden ser: visualizar un *.arff* (posteriormente se hablará en profundidad de este tipo de datos), el *package manager*, visualizar un *.sql*, ver la memoria utilizada por el programa...etc.

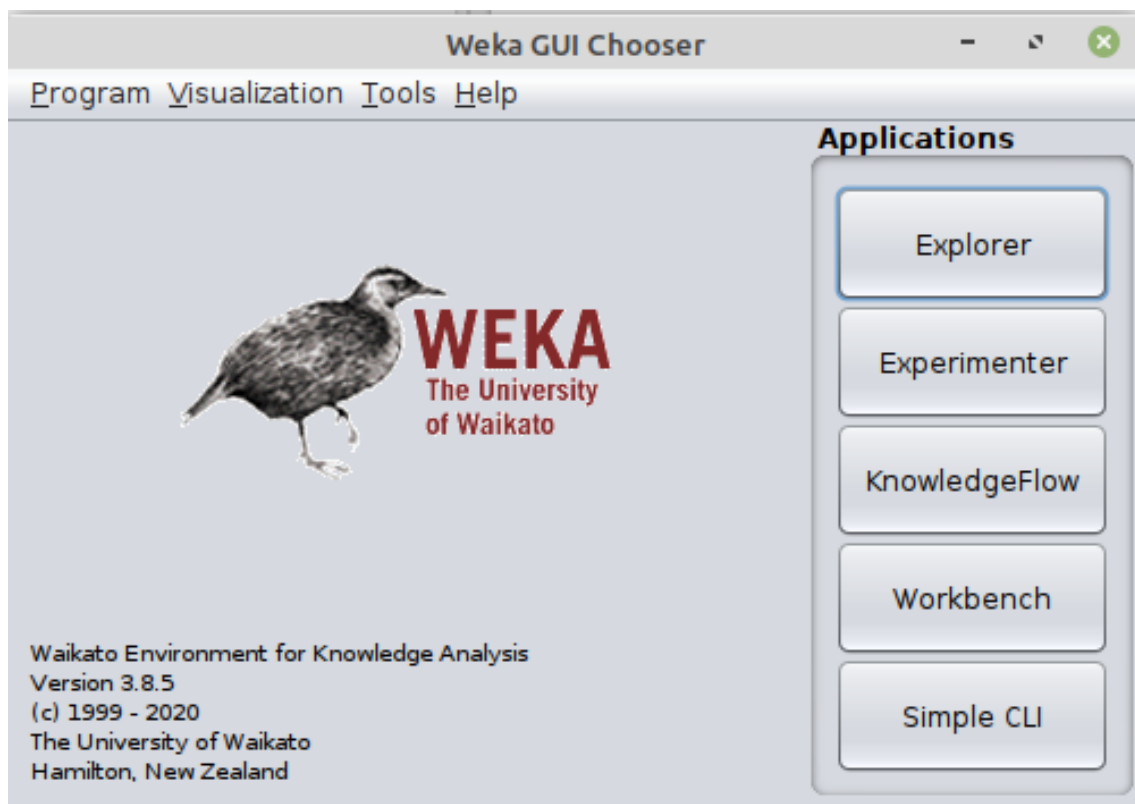


Figura 10: Interfaz gráfica de WEKA

## Explorer

La aplicación *Explorer*, es la aplicación más utilizada entre los usuarios, debido a su sencillez. La interfaz gráfica ayuda a comprender y a utilizar mejor las posibilidades que ofrece Weka. También ha sido la opción más utilizada durante el desarrollo de este proyecto, sobre todo a la hora de crear y probar el correcto funcionamiento de los algoritmos PCTRSM y PCTRF, ya que nos permite realizar diversas operaciones sobre una base de datos. Algunas de las posibilidades que nos ofrece esta aplicación son el preprocesado de datos, la clasificación, el clustering, las reglas de asociación y la visualización de datos.

En el desarrollo de este proyecto se han utilizado las pestañas de Preprocess y Classify dentro del Explorer.

- **Preprocess:** Únicamente se ha utilizado para seleccionar la base de datos sobre la que queremos aplicar el algoritmo. Esta opción nos da la posibilidad de abrir fichero con los datos que vamos a trabajar, *Open File*, y mostrándonos algunas de las características de la muestra como se aprecia en la figura 11.

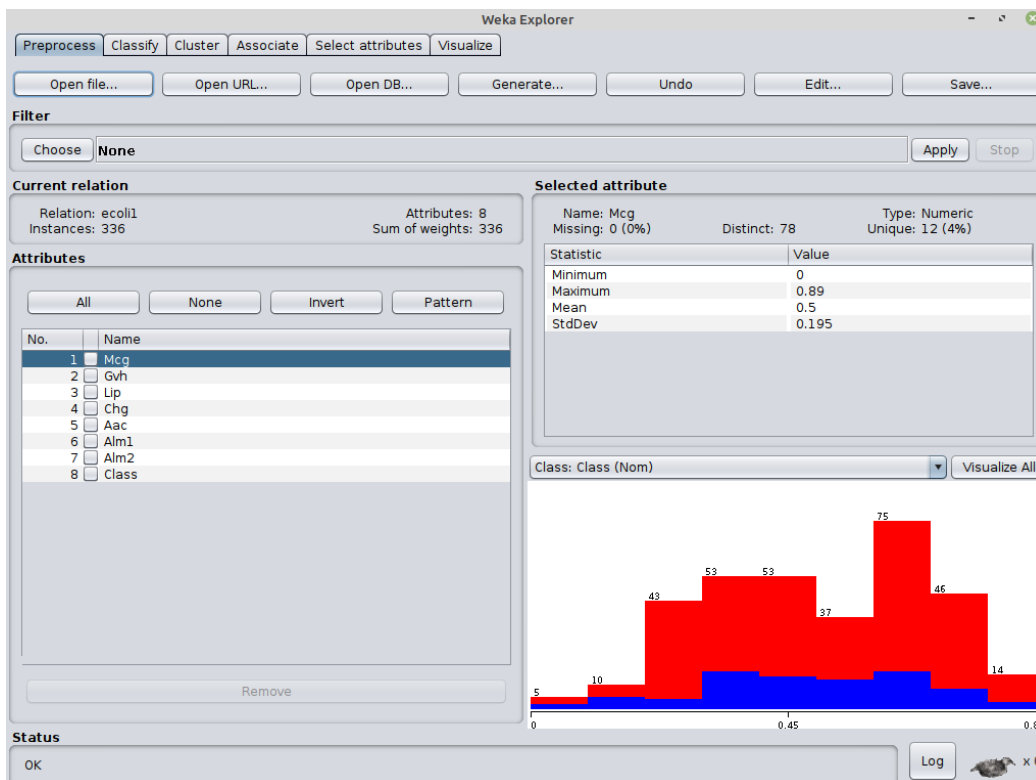


Figura 11: Interfaz gráfica del modo Explorer - Preprocess

- Classify:** La pestaña Classify se utiliza para aplicar alguno de los algoritmos de clasificación sobre la base de datos. En esta última también seleccionamos los parámetros específicos del clasificador. Una vez ejecutado el algoritmo, nos muestra los resultados de la clasificación, obteniendo valores de los criterios de bondad, como los comentados en el apartado “5.4 Criterios de Bondad”, el total de instancias clasificadas, la matriz de confusión... como se puede observar en la figura 12, dentro de la pestaña *Classifier Output*. Y en nuestro caso, al utilizar árboles de clasificación, también podemos visualizar el árbol resultante.

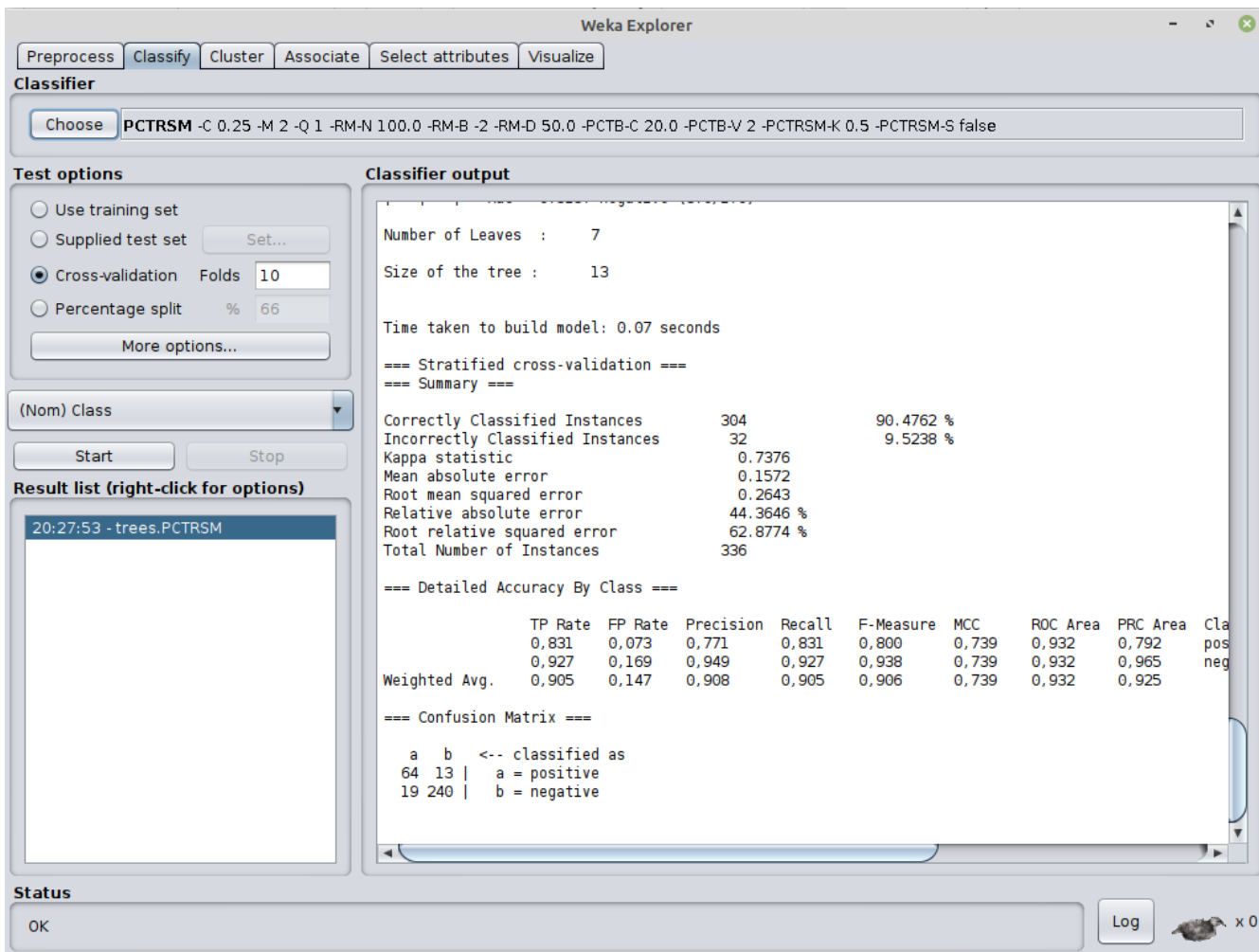


Figura 12: Interfaz gráfica del modo Explorer - Classify

## Experimenter

Este modo Experimenter, es una de las aplicaciones más útiles y efectivas que tiene Weka. Es de gran utilidad, ya que nos permite hacer experimentos costosos de forma cómoda.

Esta herramienta permite aplicar varios métodos de clasificación sobre un gran conjunto de bases de datos, como se puede observar en la figura 13. Nos permite generar un fichero con los datos y el resultado, permitiéndonos testear y compararlos bajo distintos criterios y realizar contrastes estadísticos entre ellos. Por ejemplo, el usuario puede crear un experimento en el que ejecute varios algoritmos con distintos parámetros y sobre varias bases de datos, para después analizar los resultados para determinar cuál es el mejor de ellos.

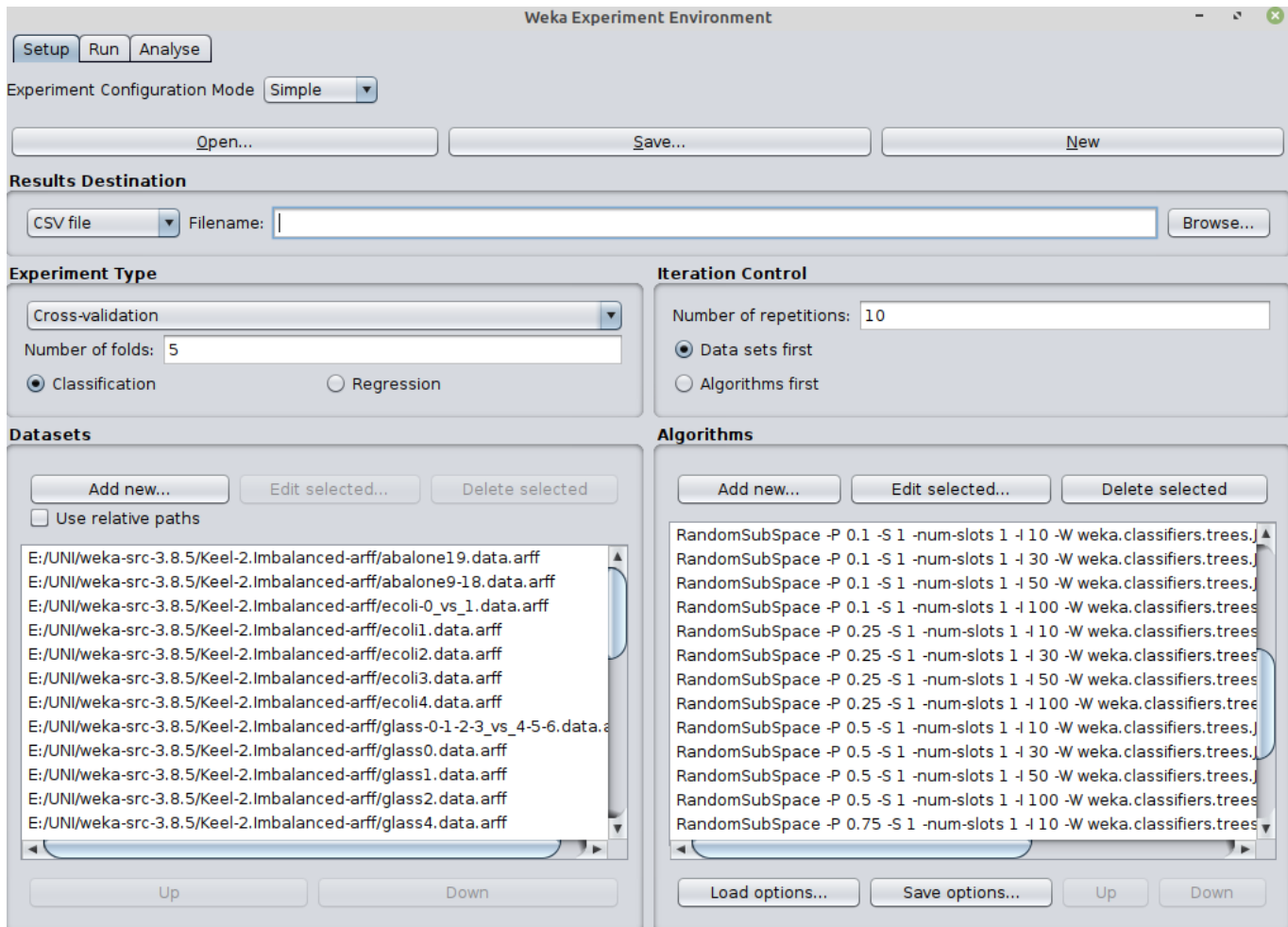


Figura 13: Interfaz gráfica Experimenter - Setup

## KnowledgeFlow

KnowledgeFlow ofrece la posibilidad de crear un “flujo” de operaciones dentro de un mismo experimento. Es decir, KnowledgeFlow presenta un “flujo” de datos, en donde el usuario puede utilizar tantos algoritmos como quiera: clasificadores, filtros, clusters...etc para crear su propio experimento.

Además presenta una interfaz que podemos ver en la figura 14, en donde puedes “arrastrar” y “soltar” el algoritmo que quieras para ponerlo en el “flujo”.

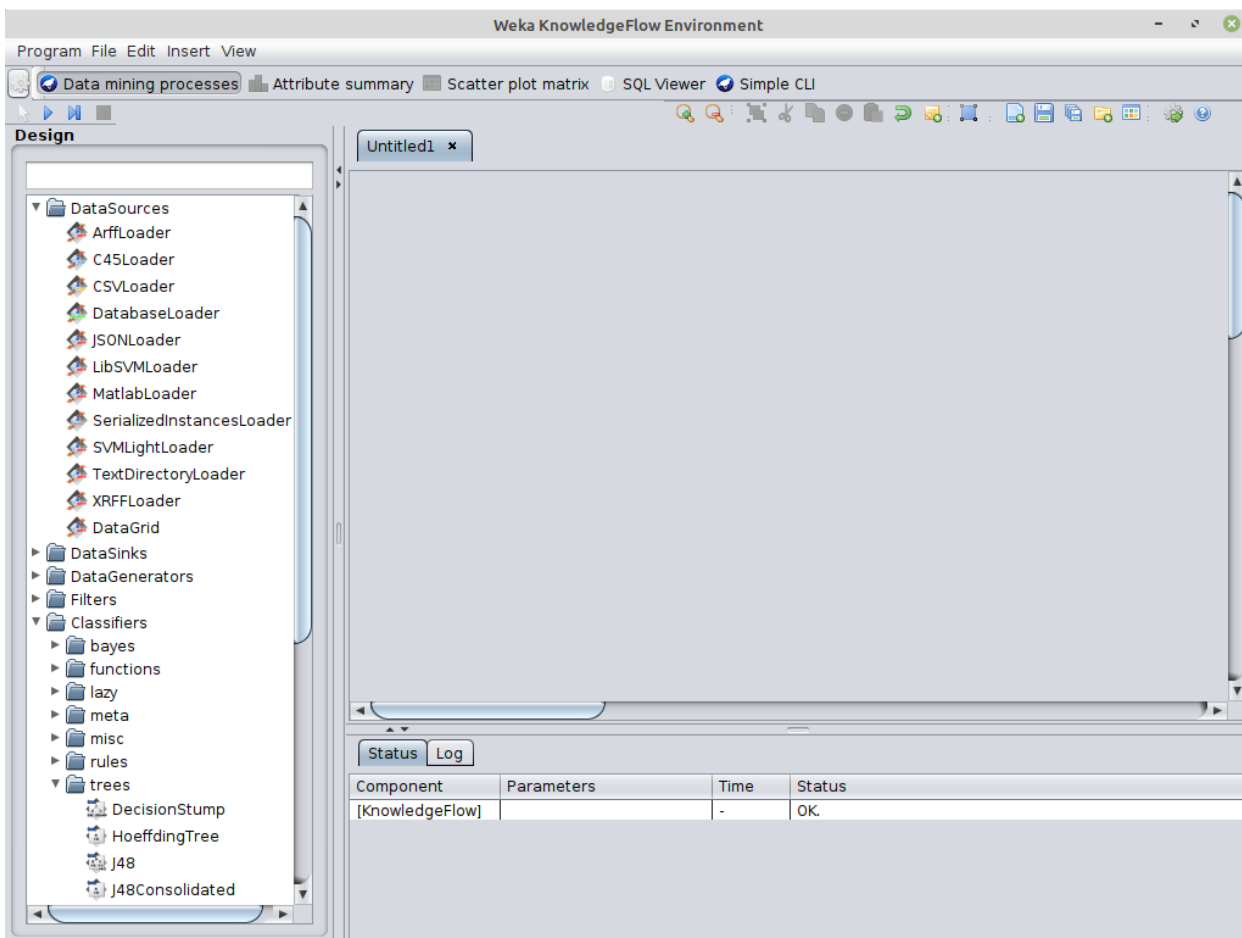


Figura 14: Interfaz gráfica del modo KnowledgeFlow

Aunque esta opción no se haya usado en este proyecto, es otra de las posibilidades que nos ofrece Weka como usuarios para realizar los experimentos.

## Simple CLI

Y por último y no por eso el menos utilizado, la simple línea de comandos. Simple CLI es una abreviación de Simple Client. Esta interfaz ofrece al usuario la posibilidad de enviar mandatos a la aplicación.

Aunque no se haya utilizado en el proyecto y la interfaz de la aplicación sea simple, es la más potente de las cinco, ya que se pueden utilizar todos los argumentos y parámetros que se quiera.

En la interfaz Explorer, por ejemplo, puede que haya opciones que no están implementadas pero que a través de la línea de comandos si se puede acceder.

Es un poco complicada de manejar ya que se deben de conocer con anterioridad los parámetros de cada algoritmo que se quiera utilizar, pero si se sabe controlar es una herramienta muy potente. Los comandos que soporta son los siguientes:

- **capabilities** <nombre de clase> <argumentos>: Enumera las capacidades de la clase especificada.
- **cls** (Clear Screen): Borra todo el contenido de la salida de la consola.
- **echo** <msg>: Escribe el mensaje por pantalla.
- **exit**: Sale de la aplicación.
- **help** <comando>: Da una descripción del comando a utilizar.
- **history**: Escribe por pantalla todos los comandos utilizados.
- **java** <nombre-de-la-clase> <argumentos>: Ejecuta el método main de la clase que se le pase como parámetro con los argumentos que se definan.
- **kill**: Elimina la tarea actual, y borra todo el contenido de la misma.
- **script** <nombre del script>: Ejecuta los comandos del script que se le indique.
- **set** [name=value]: Establece el valor de una variable. Si no se proporciona ningún par de valores name=value, se enumeran todas las variables actuales.
- **unset** <name>: Elimina una variable.

En la figura 15 se puede observar una captura de la interfaz gráfica del Simple CLI, con los comandos que ofrece.

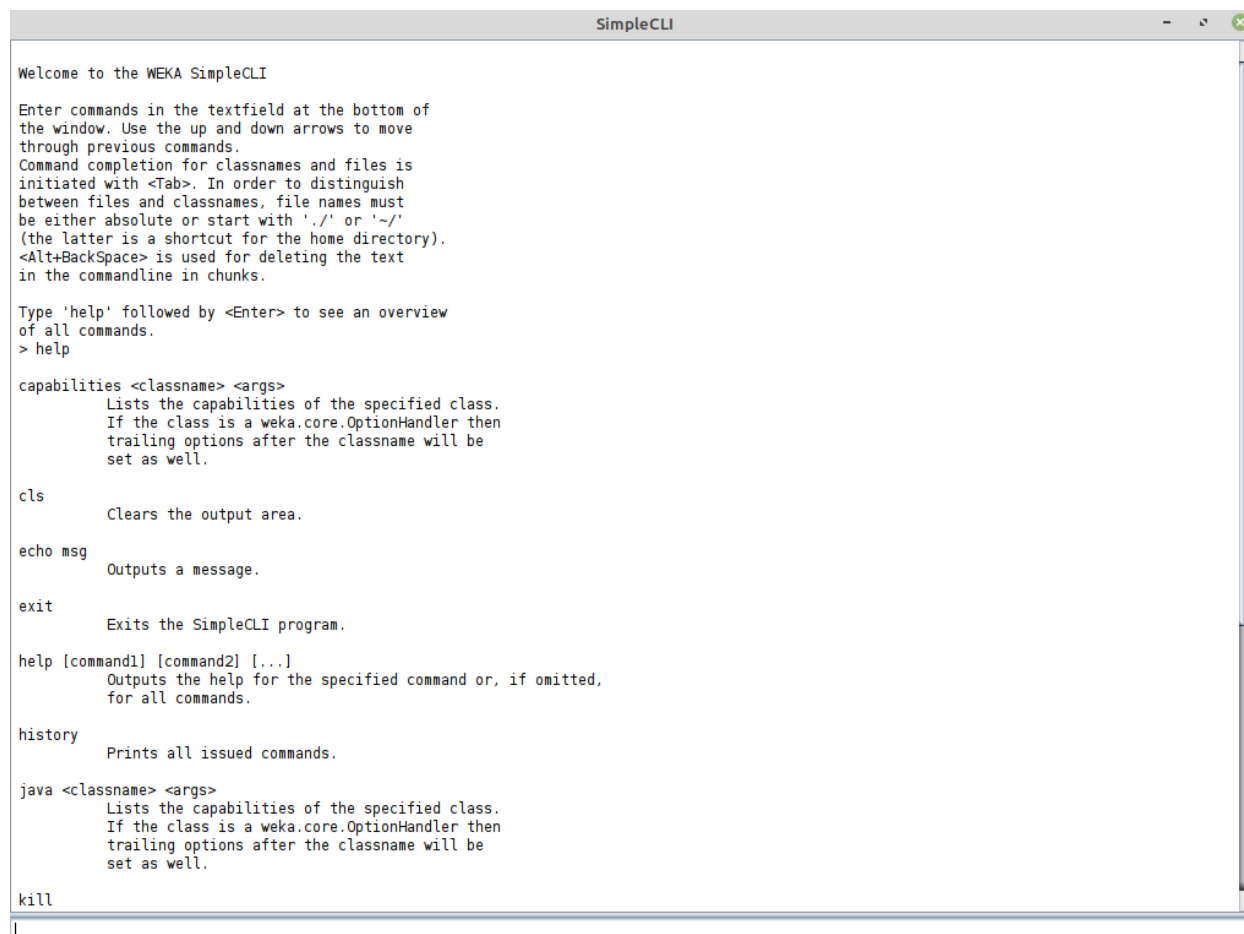


Figura 15: Interfaz gráfica del modo Simple CLI



## 6.3 DATOS

Weka por defecto trabaja con un formato de datos llamado ARFF (Attribute Relation File Format). Es un fichero ASCII que describe una lista de instancias con una serie de atributos. El archivo siempre está formado con una estructura definida en dos partes:

- **La sección de la cabecera:** En esta sección se define el nombre de la relación de la base de datos, la lista de atributos que contiene y sus tipos.
  - Nombre de relación: En la cabecera es donde se define el nombre de la relación.

**@RELATION <nombre-de-la-relacion>**

★ El <nombre-de-la-relación> es de tipo String.

- Declaración de los atributos: En este apartado se declaran los atributos que formarán parte del archivo, junto a su tipo.

**@ATTRIBUTE <nombre-del-atributo> <tipo>**

★ El <nombre-del-atributo> es de tipo String. Los tipos que acepta Weka son los siguientes:

- Numeric
- Integer - es tratado como numeric
- Real - es tratado como numeric
- Date
- String
- Relacional

- **La sección de los datos:** La sección de los datos contiene por cada línea un caso o una instancia del problema. Los datos están separados por comas.

**@ Data**

atributo1, atributo2, atributo3, ... <clase>

atributo1, atributo2, atributo3, ... <clase>

Este es un pequeño ejemplo del inicio de un archivo `.arff`, *iris.arff*, ver figura 16. Es una base de datos con 4 atributos de tipo *REAL* y 3 clases posibles:

```
@RELATION iris

@ATTRIBUTE sepallength REAL
@ATTRIBUTE sepalwidth REAL
@ATTRIBUTE petallength REAL
@ATTRIBUTE petalwidth REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
```

Figura 16: Ejemplo de archivo `.arff`

## 6.4 EXTENDIENDO WEKA

Gracias al Framework que ofrece Weka, cualquier desarrollador puede implementar su propio algoritmo e integrarlo en la aplicación. Esta es la característica que hace que Weka sea tan importante, ser un software libre con la posibilidad de modificar y añadir código en su plataforma. Además de permitir a cualquier desarrollador integrar cualquier algoritmo en la plataforma, facilitan esta práctica de forma muy clara.

Weka está implementado de manera que facilita y ayuda mucho la integración de algoritmos en su plataforma, ya que Weka dispone de una clase llamada `GenericObjectEditor`, que detecta las clases nuevas que se añaden o se eliminan. De esta forma, si un desarrollador crea una nueva clase y la coloca en el paquete específico, por ejemplo en el paquete de clasificadores, Weka detectará ese algoritmo (si está correctamente implementado) y lo hará funcionar para todas las aplicaciones (`Explorer`, `Experimenter`...etc). Gracias a esto, el desarrollador que quiera implementar algún algoritmo en Weka, no tendrá que preocuparse por el aspecto de la interfaz gráfica del mismo, ya que Weka lo hará por él.

La clave para implementar un algoritmo en esta aplicación es implementarlo tal y como Weka lo recomienda. Más adelante se explicará cómo se debe implementar un clasificador en Weka, ya que si se hace como es recomendado, el algoritmo que se implemente funcionará con todas las aplicaciones con todas sus opciones.

Como se acaba de mencionar, el código de la plataforma está dividido en los paquetes siguientes:

- **Associations:** Este paquete contiene las clases que implementan los algoritmos para realizar asociaciones.
- **AttributeSelection:** Métodos de selección de atributos.
- **Classifiers:** Paquete que contiene los algoritmos clasificadores de Weka.
- **Clusters:** Paquete que contiene las clases donde se implementa el clustering.
- **Core:** Éste es el paquete principal de Weka donde se encuentran las implementaciones de las clases principales como Attribute, Instance... y otras clases como las de Drawable, Copyable...etc.
- **Datagenerators:** Este paquete contiene las clases para generar datos y representar tests.
- **Estimators:** Estimadores de probabilidades, como *Poisson*.
- **Filters:** Paquete que contiene filtros para el preprocesado para aplicar a los datos.
- **Gui:** Este paquete contiene las clases que forman la interfaz gráfica de la plataforma.

#### 6.4.1 IMPLEMENTACIÓN

Para implementar el algoritmo correctamente en WEKA, hay una serie de métodos que han de ser implementados obligatoriamente en la nueva clase principal que se cree. Dependiendo de si el algoritmo a implementar es un clasificador o meta-clasificador, se implementarán unos u otros métodos [7].

Hay una serie de métodos que son comunes en todos ellos. Para entender mejor estos métodos, se adjuntan unas figuras de las implementaciones de los métodos para el algoritmo J48:

- **globalInfo():** Este método permite mostrar una breve descripción del algoritmo en la interfaz gráfica, más concretamente en las aplicaciones Explorer y Experimenter.

```

/**
 * Returns a string describing classifier
 *
 * @return a description suitable for displaying in the explorer/experimenter
 *         gui
 */
public String globalInfo() {

    return "Class for generating a pruned or unpruned C4.5 decision tree. For more "
        + "information, see\n\n" + getTechnicalInformation().toString();
}

```

Figura 17: Método globalInfo de J48

- **listOptions()**: Devuelve la enumeración de opciones y parámetros posibles del algoritmo, que nos sirve en modo de ayuda en la aplicación de línea de comandos.

```

@Override
public Enumeration<Option> listOptions() {

    Vector<Option> newVector = new Vector<Option>(13);

    newVector.addElement(new Option("\tUse unpruned tree.", "U", 0, "-U"));
    newVector.addElement(new Option("\tDo not collapse tree.", "O", 0, "-O"));
    newVector.addElement(new Option("\tSet confidence threshold for pruning.\n"
        + "\t(default 0.25)", "C", 1, "-C <pruning confidence>"));
    newVector.addElement(new Option(
        "\tSet minimum number of instances per leaf.\n" + "\t(default 2)", "M",
        1, "-M <minimum number of instances>"));

    ...

    newVector.addAll(Collections.list(super.listOptions()));

    return newVector.elements();
}

```

Figura 18: Método listOptions de J48

- **setOptions(String[])**: Este método se encarga de analizar y tratar los parámetros del algoritmo, metiéndolos en variables globales, para facilitar su uso. Tendrá también valores por defecto para las variables en caso de no introducirse ningún valor por el usuario.

```

@Override
public void setOptions(String[] options) throws Exception {

    // Other options
    String minNumString = Utils.getOption('M', options);
    if (minNumString.length() != 0) {
        m_minNumObj = Integer.parseInt(minNumString);
    } else {
        m_minNumObj = 2;
    }
    m_binarySplits = Utils.getFlag('B', options);
    m_useLaplace = Utils.getFlag('A', options);
    m_useMDLcorrection = !Utils.getFlag('J', options);

    ...

    super.setOptions(options);
}

```

Figura 19: Método setOptions de J48

- **getOptions()**: Esta función devuelve un array de strings con las opciones o la configuración que compone el clasificador actual. Las opciones deben coincidir con las fijadas en el método setOptions.

```

@Override
public String[] getOptions() {

    Vector<String> options = new Vector<String>();

    // Issue some warnings for the current configuration if necessary
    if (m_unpruned) {
        if (!m_subtreeRaising) {
            System.err.println("WARNING: Subtree raising does not need to be unset for an unpruned tree!");
        }
        if (m_reducedErrorPruning) {
            System.err.println("WARNING: Unpruned tree and reduced error pruning cannot be selected simultaneously!");
        }
    }
    if (m_unpruned || m_reducedErrorPruning) {
        if (m_CF != 0.25f) {
            System.err.println("WARNING: Changing the confidence will only affect error-based pruning!");
        }
    }
    if (m_unpruned || !m_reducedErrorPruning) {
        if (m_Seed != 1) {
            System.err.println("WARNING: Changing the seed only makes sense when using reduced error pruning");
        }
        if (m_numFolds != 3) {
            System.err.println("WARNING: Changing the number of folds does not make sense if " +
                "reduced error pruning is not selected.");
        }
    }

    if (m_noCleanup) {
        options.add("-L");
    }
    if (!m_collapseTree) {
        options.add("-O");
    }

    ...

    Collections.addAll(options, super.getOptions());

    return options.toArray(new String[0]);
}

```

Figura 20: Método getOptions de J48

- **getCapabilities():** Devuelve la información de qué tipo de datos (atributos, clases, instancias...) maneja el clasificador. Es necesario dejar claro qué datos puede manejar el algoritmo, ya que posteriormente no se podrá hacer esa comprobación.

```
/**
 * Returns default capabilities of the classifier.
 *
 * @return the capabilities of this classifier
 */
@Override
public Capabilities getCapabilities() {
    Capabilities result;

    result = new Capabilities(this);
    result.disableAll();
    // attributes
    result.enable(Capability.NOMINAL_ATTRIBUTES);
    result.enable(Capability.NUMERIC_ATTRIBUTES);
    result.enable(Capability.DATE_ATTRIBUTES);
    result.enable(Capability.MISSING_VALUES);

    // class
    result.enable(Capability.NOMINAL_CLASS);
    result.enable(Capability.MISSING_CLASS_VALUES);

    // instances
    result.setMinimumNumberInstances(0);

    return result;
}
```

Figura 21: Método getCapabilities de J48

- **buildClassifier(Instances):** Este es el método más importante de todos, ya que es el encargado de generar el modelo del clasificador. No hay un estilo fijo para implementar este método, ya que depende mucho de qué tipo de clasificador se esté realizando. Weka recomienda guiarse y usar como referencia algoritmos semejantes y poco a poco, adaptándolo a nuestras necesidades, escribir el código.

```

@Override
public void buildClassifier(Instances instances) throws Exception {

    if ((m_unpruned) && (!m_subtreeRaising)) {
        throw new Exception("Subtree raising does not need to be unset for unpruned trees!");
    }
    if ((m_unpruned) && (m_reducedErrorPruning)) {
        throw new Exception("Unpruned tree and reduced error pruning cannot be selected simultaneously!");
    }
    if ((m_unpruned) && (m_CF != 0.25f)) {
        throw new Exception("It does not make sense to change the confidence for an unpruned tree!");
    }
    if ((m_reducedErrorPruning) && (m_CF != 0.25f)) {
        throw new Exception("Changing the confidence does not make sense for reduced error pruning.");
    }
    if (!(m_reducedErrorPruning) && (m_numFolds != 3)) {
        throw new Exception("Changing the number of folds does not make sense if"
            + " reduced error pruning is not selected.");
    }
    if (!(m_reducedErrorPruning) && (m_Seed != 1)) {
        throw new Exception("Changing the seed does not make sense if"
            + " reduced error pruning is not selected.");
    }
    if ((m_CF <= 0) || (m_CF >= 1)) {
        throw new Exception("Confidence has to be greater than zero and smaller than one!");
    }
    getCapabilities().testWithFail(instances);

    ModelSelection modSelection;

    if (m_binarySplits) {
        modSelection = new BinC45ModelSelection(m_minNumObj, instances,
            m_useMDLcorrection, m_doNotMakeSplitPointActualValue);
    }
    . . .
}

```

Figura 22: Método buildClassifier de J48

- **toString():** Este método sirve para exportar el modelo construido. Este método no es obligatorio de implementar, pero ayuda al usuario a ver las propiedades del modelo.

```

@Override
public String toString() {

    if (m_root == null) {
        return "No classifier built";
    }
    if (m_unpruned) {
        return "J48 unpruned tree\n-----\n" + m_root.toString();
    } else {
        return "J48 pruned tree\n-----\n" + m_root.toString();
    }
}

```

Figura 23: Método toString de J48

- **distributionForInstance(Instance)**: Otro de los métodos más importantes. Devuelve un array de probabilidades de la predicción de la instancia dada.

```

/**
 * Returns class probabilities for an instance.
 *
 * @param instance the instance to calculate the class probabilities for
 * @return the class probabilities
 * @throws Exception if distribution can't be computed successfully
 */
@Override
public double[] distributionForInstance(Instance instance)
    throws Exception {

    return m_root.distributionForInstance(instance, m_useLaplace);
}

```

Figura 24: Método `distributionForInstance` de J48

- **classifyInstance(Instance)**: Devuelve la clasificación del objeto *Instance* dado. En caso de ser nominal, se devolverá un índice de la clase predicha.

```

/**
 * Classifies an instance.
 *
 * @param instance the instance to classify
 * @return the classification for the instance
 * @throws Exception if instance can't be classified successfully
 */
@Override
public double classifyInstance(Instance instance) throws Exception {

    return m_root.classifyInstance(instance);
}

```

Figura 25: Método `classifyInstance` de J48

- **main(String[])**: Ejecuta el clasificador con los parámetros que se le pasen, con la llamada al método *runClassifier*, que se encargará de las llamadas y errores que puedan ocurrir durante la ejecución del algoritmo.



```

/**
 * Main method for testing this class
 *
 * @param argv the commandline options
 */
public static void main(String[] argv) {
    runClassifier(new J48(), argv);
}
}

```

Figura 26: Método main de J48

Hasta aquí serían los métodos necesarios para implementar un clasificar. Si el algoritmo que se quiere introducir en Weka es un meta-clasificador, se tendrían que implementar las siguientes funciones, dependiendo si es un *SingleClassifierEnhancer* o un *MultipleClassifierCombiner*.

#### *SingleClassifierEnhancer:*

- **defaultClassifierString():** devuelve el nombre de la clase del clasificador que se está utilizando por defecto en el meta-clasificador.
- **setClassifier(Classifier):** dado un clasificador como parámetro, pone ese clasificador como base para el meta-clasificador.
- **getClassifier():** devuelve el clasificador base actual del meta-clasificador.

#### *MultipleClassifierEnhancer:*

- **setClassifiers(Classifiers[]):** Introduce como clasificadores base los elementos del array.
- **getClassifiers():** devuelve un array con los clasificadores base del meta clasificador.
- **getClassifier(int):** devuelve el clasificador en la posición dada como parámetro.

Tras haber implementado los métodos que Weka obliga a implementar, hay una serie de directrices que son interesantes a la hora de implementar un nuevo algoritmo en Weka. Para este proyecto, se han implementado los métodos necesarios para que los parámetros que se añadan a nuestro algoritmo sean visibles desde la interfaz gráfica.

En primer lugar, se puede ver en la figura 27 un ejemplo de la interfaz gráfica para la elección de los parámetros del algoritmo PCTRSM.

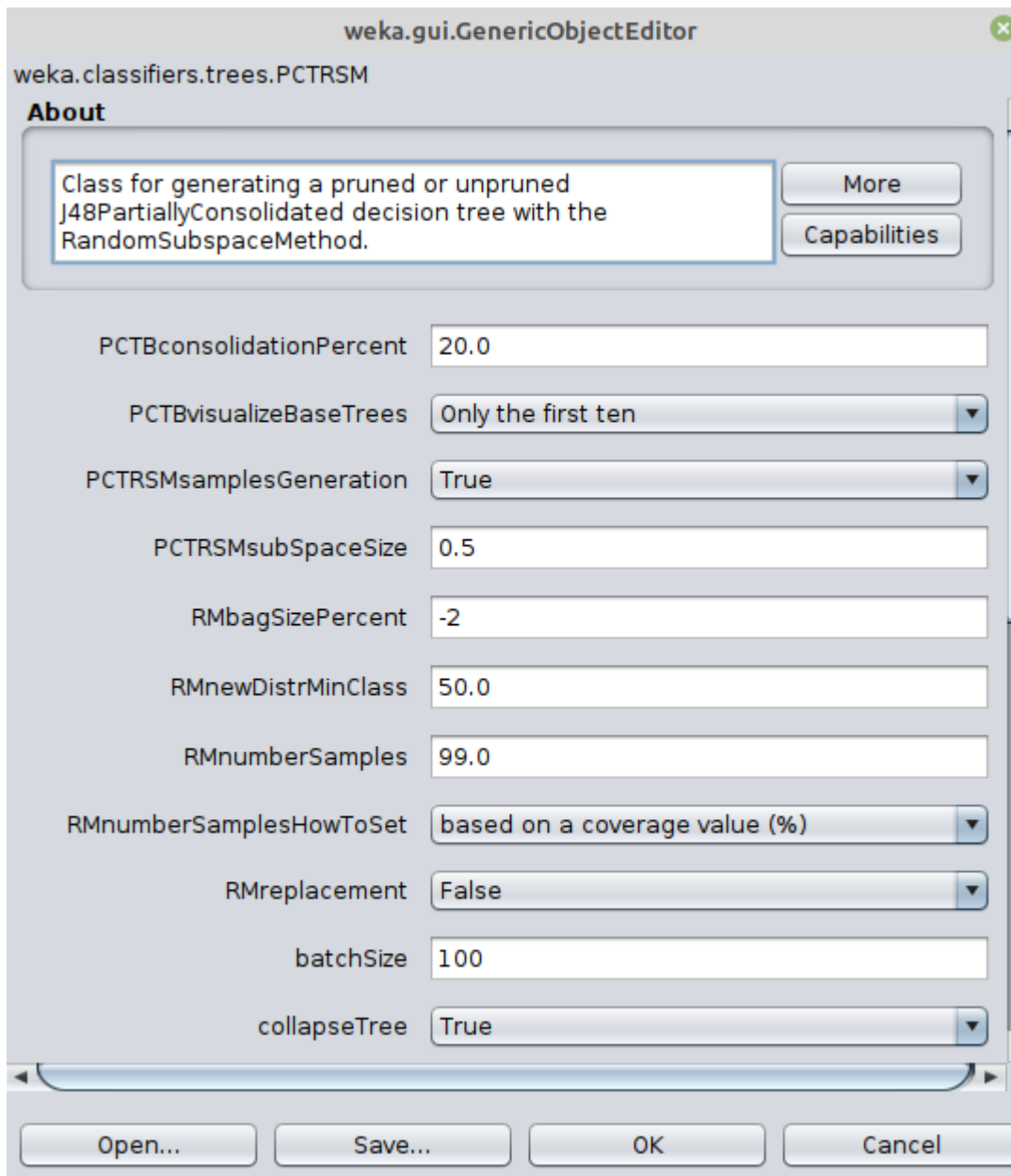


Figura 27: Interfaz gráfica para la elección de parámetros

Por cada uno de estos nuevos parámetros que se quieran añadir, habrá que implementar tres funciones en nuestro código para que Weka sea capaz de mostrarlos por la interfaz gráfica y tomar los valores que posteriormente se introduzcan.

- **public String <NombreDelParámetro>TipText():** sirve para mostrar una pequeña descripción del parámetro. Esta descripción será visualizada a través de la interfaz gráfica.
- **public <Tipo> get<NombreDelParámetro>():** devuelve el valor del parámetro.

- **public set<NombreDelParámetro>(<valor>):** cambia el valor de la variable del parámetro.

A modo de ejemplo, se usará el parámetro *PCTBconsolidationPercent* del algoritmo *J48PartiallyConsolidated*. Si queremos que el parámetro *PCTBconsolidationPercent* aparezca en la interfaz gráfica de Weka, deberemos implementar los métodos antes mencionados de la manera que se ve en las figuras 28, 29 y 30, y así sería visible en la interfaz.

```
/**
 * Returns the tip text for this property
 * @return tip text for this property suitable for
 * displaying in the explorer/experimenter gui
 */
public String PCTBconsolidationPercentTipText() {
    return "Consolidation percent for use after the consolidation process";
}
```

Figura 28: PCTBconsolidationPercentTipText

```
/**
 * Get the value of ConsolidationPercent.
 *
 * @return Value of ConsolidationPercent.
 */
public float getPCTBconsolidationPercent() {

    return m_PCTBconsolidationPercent;
}
```

Figura 29: getPCTBconsolidationPercent

```
/**
 * Set the value of ConsolidationPercent.
 *
 * @param v Value to assign to ConsolidationPercent.
 * @throws Exception if an option is not supported
 */
public void setPCTBconsolidationPercent(float v) throws Exception {
    if ((v < 0) || (v > 100))
        throw new Exception("The consolidation percent (%) has to "
            + "be a value greater than or equal to zero and smaller " +
            "than or equal to 100!");
    m_PCTBconsolidationPercent = v;
}
```

Figura 30: setPCTBconsolidationPercent



## 7. PCT-ENSEMBLES

Como se ha explicado previamente en el apartado 5.3, *CLASIFICADORES MÚLTIPLES Ó ENSEMBLES*, el PCTBagging es un clasificador múltiple basado en la construcción de árboles parcialmente consolidados. Este es el algoritmo en el que están basados el PCTRSM y el PCTRF. A su vez, al estar el PCTBagging basado en el CTC y el C4.5, estos también serán la base para nuestros algoritmos.

Tanto el PCTRSM como el PCTRF tienen como objetivo añadir la aleatoriedad a esta construcción de árboles parcialmente consolidados y conocer si ésta mejora, iguala o empeora la clasificación de los casos.

Ambos algoritmos están basados en el PCTBagging, es decir, se construye un árbol parcialmente consolidado con un conjunto de muestras, como pueden ser muestras bootstrap, balanceadas, sin remuestreo (Sólo este último motivado por la selección de atributos del RSM o RF puros) ... Y a partir de ahí, se construye un Bagging con el conjunto de árboles. La diferencia radica en la construcción de los árboles individuales, tanto durante el proceso de consolidación parcial como posteriormente, cuando se desarrollan los árboles independientemente como en Bagging, y es que estos árboles se construyen bajo el mismo criterio que en RSM o RF.

El algoritmo PCTRSM introduce la aleatoriedad en este proceso con la generación de los subespacios de atributos. A cada una de las muestras generadas se le asigna un número de atributos, que puede variar desde un único atributo hasta el 100% de ellos, con los que realiza todos los procesos posteriores, que son la creación del árbol consolidado y los árboles parcialmente consolidados. Las variables que hayan sido descartadas solamente pueden aparecer en el árbol consolidado, debido a la votación que se realiza, siendo el posible ganador una variable que no está incluida en alguno de los subespacios.

El algoritmo PCTRF sigue la misma idea que el PCTRSM, con la diferencia de cuando se usan esos subespacios. La generación de este espacio de atributos reducido se realiza de la misma manera, eligiendo un valor entre una única variable y el 100% de ellas, pero en este caso, se llevará a cabo este proceso para cada nodo de cada árbol en vez de asignar un subespacio a cada muestra.

## 8. IMPLEMENTACIÓN DE PCTRSM Y PCTRF

En este capítulo del proyecto se va a explicar cómo se ha logrado implementar los algoritmos PCTRSM y PCTRF en Weka. Para ello, se explicarán los argumentos y los parámetros añadidos a la versión del J48PartiallyConsolidated, así como los nuevos métodos o modificaciones que hayan llegado a realizar.

### 8.1 PCTRSM

En este apartado se van a explicar las clases, parámetros y métodos que han sido necesarios para diseñar e implementar el código del PCTRSM en WEKA.

Para el correcto funcionamiento del algoritmo, se ha necesitado crear una clase principal o base que se llama PCTRSM, como el nombre del algoritmo, y 4 clases secundarias nuevas para poder aplicar las bases del *RSM*, los subespacios generados a la hora de crear los árboles para cada submuestra. Estas 4 clases estarán incluidas en el paquete *Classifiers.Trees.PCTRSM*.

- **Clase Base: PCTRSM**

El nombre que se le ha dado a la clase principal ha sido PCTRSM, el mismo nombre que al algoritmo. Se ha escogido como superclase el J48PartiallyConsolidated y se han añadido los dos parámetros nuevos que serán necesarios para el desarrollo del código. Esta clase pertenece al paquete de *classifiers.trees* de weka, al igual que el J48, el J48Consolidated y el J48PartiallyConsolidated.

```
package weka.classifiers.trees;

import java.util.ArrayList;

public class PCTRSM extends J48PartiallyConsolidated implements OptionHandler, Drawable, Matchable, Sourcable,
    WeightedInstancesHandler, Summarizable, AdditionalMeasureProducer, TechnicalInformationHandler {

    /** for serialization */
    private static final long serialVersionUID = 6167369031857640703L;

    /** Size of each subSpace: if less than 1 as a percentage of the number of attributes, otherwise
     * the absolute number of attributes
     * Based on weka.classifiers.trees.RandomTree.m KValue, if 0,  $\text{int}(\log_2(\#\text{predictors}) + 1)$  is used. */
    protected float m_PCTRSMSubSpaceSize = (float) 0.5;

    /** Boolean to choose sample generation or not*/
    protected boolean m_PCTRSMSamplesGeneration = true;
}
```

Figura 31: Definición de la clase y sus atributos - PCTRSM

Como se puede ver en la figura 31, los dos parámetros añadidos han sido los siguientes:

- **m\_PCTRSMsubSpaceSize:** Este valor determina el tamaño del subespacio de los atributos que se genera para cada submuestra. Tiene 3 tipos de valores distintos que puede aceptar, como combinación de la implementación en WEKA de los algoritmos RSM y RF:
  - **Valor 0:** Esta es una opción implementada debido al Random Forest, es una alternativa que se añadió tras la realización del PCTRF. Random Forest, por defecto, si el valor para el nuevo espacio de atributos es menor que 1, o es igual a 0, utilizará la función  $\text{int}(\log_2(\#predictors) + 1)$ , siendo *predictors* el número de atributos de los datos. Para nuestro algoritmo, se utilizará el valor de 0.
  - **Valor <1:** En caso de utilizar un valor decimal, un valor menor que 1 y mayor que 0, se interpretará como el porcentaje de atributos del espacio original a mantener, pero en tanto por uno. Por defecto m\_PCTRSMsubSpaceSize tiene un valor de 0.5, igual que el algoritmo RSM, que en WEKA es el RandomSubSpace (su implementación), para indicar que se seleccionará el 50% de los atributos existentes.
  - **Valor >1:** Si el valor que se utiliza es mayor que 1, será el total o valor absoluto de atributos a mantener, sin importar el espacio de atributos original. Si se indica un valor mayor al total de atributos, se mantendrá el total de los mismos.

Los métodos utilizados para implementar este parámetro son los siguientes, como se puede ver en las figuras 32, 33 y 34:

```
/**
 * Returns the tip text for this property
 * @return tip text for this property suitable for
 * displaying in the explorer/experimenter gui
 */
public String PCTRSMsubSpaceSizeTipText() {
    return "Size of each subSpace:\n"
        + "If 0,  $\text{int}(\log_2(\#predictors) + 1)$  is used.\n"
        + "If less than 1 as a percentage of the number of attributes,\n"
        + "otherwise the absolute number of attributes.";
}
```

Figura 32: Método PCTRSMsubSpaceSizeTipText

```

/**
 * Get the value of subSpaceSize.
 *
 * @return Value of subSpaceSize.
 */
public float getPCTRSMsubSpaceSize() {
    return m_PCTRSMsubSpaceSize;
}

```

Figura 33: Método getPCTRSMsubSpaceSize

```

/**
 * Set the value of subSpaceSize.
 *
 * @param v Value to assign to subSpaceSize.
 */
public void setPCTRSMsubSpaceSize(float v) throws Exception {
    m_PCTRSMsubSpaceSize = v;
}

```

Figura 34: Método setPCTRSMsubSpaceSize

- **m\_PCTRSMsamplesGeneration:** Con este valor se determinará si va a realizar una generación de muestras heredada del J48Consolidated o se mantiene el estado original de los datos como en el RandomSubSpace.
  - **True:** Valor por defecto del clasificador. Si se elige este método, se realizará una generación de muestras.
  - **False:** En caso de no elegir una generación de muestras, la base de datos original no será modificada y todas las submuestras serán idénticas.

Los métodos utilizados para implementar este parámetro son los siguientes, como se puede observar en las figuras 35, 36 y 37:



```

/**
 * Returns the tip text for this property
 * @return tip text for this property suitable for
 * displaying in the explorer/experimenter gui
 */
public String PCTRSMsamplesGenerationTipText() {
    return "Samples generation mode. true for j48PartiallyConsolidated samples generation type. "
        + "If false is selected, RMnumberSamples" + "will be the number of samples to generate ";
}

```

Figura 35: Método PCTRSMsamplesGenerationTipText

```

/**
 * Get the value of samplesGeneration.
 *
 * @return Value of samplesGeneration.
 */
public boolean getPCTRSMsamplesGeneration() {
    return m_PCTRSMsamplesGeneration;
}

```

Figura 36: Método getPCTRSMsamplesGeneration

```

/**
 * Set the value of samplesGeneration.
 *
 * @param samplesGeneration Value to assign to samplesGeneration.
 */
public void setPCTRSMsamplesGeneration(boolean samplesGeneration) throws Exception {
    if (!samplesGeneration) {
        setRMnumberSamplesHowToSet(new SelectedTag(NumberSamples_FixedValue, TAGS_WAYS_TO_SET_NUMBER_SAMPLES));
        RandomSubSpace rss = new RandomSubSpace();
        setRMnumberSamples((float) rss.getNumIterations());
    }
}
/**
 * If samplesGeneration is true the user can choose to set the number of samples
 * by 'using a fixed value' or 'based on a coverage value (%)'.
 */
m_PCTRSMsamplesGeneration = samplesGeneration;
}

```

Figura 37: Método setPCTRSMsamplesGeneration

## Métodos principales

Como se ha comentado previamente, para que Weka detecte y ejecute nuestros algoritmos, es necesaria la implementación de varios métodos obligatorios para que así pueda reconocer e incluir el clasificador en las diferentes aplicaciones. Hay algunos métodos de estos explicados anteriormente, como *distributionForInstance*, *classifyInstance*... que no se han implementado, ya que funcionan de la misma manera que los métodos que ha heredado ya sean de J48, J48Consolidated o J48PartiallyConsolidated.

### listOptions

Con este método, figura 38, se devuelven las opciones del clasificador. A parte de las opciones fijadas por el J48, el J48Consolidated y el J48PartiallyConsolidated, se añaden los dos nuevos parámetros posibles para este algoritmo.

```
public Enumeration<Option> listOptions() {
    Vector<Option> newVector = new Vector<Option>();

    // J48, J48Consolidated and J48PartiallyConsolidated options
    // =====
    Enumeration<Option> en;
    en = super.listOptions();
    while (en.hasMoreElements())
        newVector.addElement((Option) en.nextElement());

    // PCTRSM options
    // =====
    newVector.addElement(new Option(
        "\tSize of each subspace:\n" + "\t(default 0.5)"
        + "\t\t< 1: percentage of the number of attributes\n"
        + "\t\t>=1: absolute number of attributes\n"
        + "\t\t==0: int(log 2(#predictors) + 1)\n",
        "PCTRSM-K", 1, "-PCTRSM-K <subSpace size>"));

    newVector.addElement(new Option("\tSamples generation mode\n" + "\t(default true)", "PCTRSM-S", 1,
        "-PCTRSM-S <Samples generation>"));

    return newVector.elements();
}
```

Figura 38: Método listOptions - PCTRSM

### setOptions

Esta función se encarga de guardar los valores de las opciones o parámetros del clasificador. También se encarga de tratar excepciones en relación a estas asignaciones, restringiendo qué valores pueden tomar.

```

public void setOptions(String[] options) throws Exception {

    // PCTRSM options
    // =====
    String PCTRSMattributesPercent = Utils.getOption("PCTRSM-K", options);
    if (PCTRSMattributesPercent.length() != 0) {
        setPCTRSMsubSpaceSize(Float.valueOf(PCTRSMattributesPercent));
    } else {
        setPCTRSMsubSpaceSize((float) 0.5);
    }

    String PCTRSMsamplesGeneration = Utils.getOption("PCTRSM-S", options);
    if (PCTRSMsamplesGeneration.length() != 0) {
        setPCTRSMsamplesGeneration(Boolean.valueOf(PCTRSMsamplesGeneration));
    } else {
        setPCTRSMsamplesGeneration(true);
    }

    // J48, J48Consolidated and J48PartiallyConsolidated options
    // =====
    super.setOptions(options);
}

```

Figura 39: Método setOptions - PCTRSM

## getOptions

Esta función devuelve un array de strings con las opciones o la configuración que compone el clasificador (según el formato de la línea de comandos). Las opciones deben coincidir con las fijadas en el método setOptions.

```

public String[] getOptions() {

    Vector<String> result = new Vector<String>();

    // J48, J48Consolidated, J48PartiallyConsolidated options
    // =====
    String[] options = super.getOptions();
    for (int i = 0; i < options.length; i++)
        result.add(options[i]);

    // PCTRSM options
    // =====
    result.add("-PCTRSM-K");
    result.add("" + m_PCTRSMsubSpaceSize);

    result.add("-PCTRSM-S");
    result.add("" + m_PCTRSMsamplesGeneration);

    return (String[]) result.toArray(new String[result.size()]);
}

```

Figura 40: Método getOptions - PCTRSM

## globalInfo

Con el método global info se devuelve una pequeña descripción general del algoritmo.

```
public String globalInfo() {
    return "Class for generating a pruned or unpruned J48PartiallyConsolidated "
        + "decision tree with the RandomSubspaceMethod"
        + ". For more information, see\n\n" + getTechnicalInformation().toString();
}
```

Figura 41: Método globalInfo - PCTRSM

## buildClassifier

Este método es la función principal de esta clase. Es el encargado de generar el modelo de nuestro clasificador.

Como se ve en la figura 42, comienza comprobando si el algoritmo es capaz de procesar los datos. Si esto es posible, eliminará los casos sin clase definida y continuará creando el vector de submuestras.

```
public void buildClassifier(Instances instances) throws Exception {

    // can classifier tree handle the data?
    getCapabilities().testWithFail(instances);

    // remove instances with missing class before generate samples
    instances = new Instances(instances);
    instances.deleteWithMissingClass();
}
```

Figura 42: Primer segmento método buildClassifier - PCTRSM

En función a la opción que se haya elegido para el parámetro `m_PCTRSMsamplesGeneration`, ver figura 43, se realizará una generación de muestras, al igual que se realiza en el algoritmo `J48Consolidated`, o se llamará a la función `generatePCTRSMSamples`, que se explicará en el siguiente apartado.

```

// Generate as many samples as the number of samples with the given instances
Instances[] samplesVector;

if (m_PCTRSMsamplesGeneration) { // if PCTRSMsamplesGeneration is true, regular samples generation of
    // j48Consolidated
    samplesVector = generateSamples(instances);
} else { // if PCTRSMsamplesGeneration is false, no sampling generation, RSM regular mode
    samplesVector = generateRSMSamples(instances);
}

```

Figura 43: Segundo segmento método buildClassifier - PCTRSM

Como se puede observar en la figura 44, con el vector de submuestras ya creado, se necesita obtener una serie de valores para generar los subespacios que se asignará a cada una de las muestras. Los cuatro valores que se necesitan son:

- *Array* con los índices del espacio original de atributos: ***attIndexVector***
- El tamaño del subespacio, es decir, el número de atributos a mantener: ***subSpaceSize*** Este dato se obtiene mediante la función `numberOfAttributes`, que se explicará posteriormente.
- Un objeto *Random* generado a partir de la *seed* fijada, para generar los subespacios de manera aleatoria
- Por último, un *ArrayList* de *arrays* que contendrá todos los subespacios generados: ***localAttIndexSubSpaceVectorBySample***.

```

// Array of the attributes indices// {0,1,2,3,4...}
// (based on weka.classifiers.meta.RandomSubSpace.buildClassifier(Instances))
Integer[] attIndexVector = new Integer[instances.numAttributes() - 1];
Integer[] indices;
int classIndex = instances.classIndex();
int offset = 0;
for (int i = 0; i < attIndexVector.length; i++) {
    if (i != classIndex) {
        attIndexVector[offset++] = i;
    }
}

// Number of attributes to keep
int subSpaceSize = numberOfAttributes(attIndexVector.length, m_PCTRSMsubSpaceSize);

// Generate a random object with the seed
Random random = instances.getRandomNumberGenerator(m_Seed);

// ArrayList to save the arrays of the attributes of each sample to keep
ArrayList<Integer[]> localAttIndexSubSpaceVectorBySample = new ArrayList<Integer[]>();

```

Figura 44: Tercer segmento método buildClassifier - PCTRSM

Cuando todas las variables necesarias para la generación de los espacios estén definidas, se procederá a su creación.

Como se puede ver en la figura 45, con un bucle que irá desde 0 hasta la longitud del vector de muestras, se realizarán los siguientes pasos:

- Se crea una copia del vector *attIndexVector*.
- Se ejecutará el método Shuffle sobre dicha copia de *attIndexVector*. La función shuffle toma una lista dada por el usuario y la baraja aleatoriamente.
- Tras mezclar los índices de los atributos, nos quedaremos con los primeros valores del array que nos indique *subSpaceSize*, el número de atributos a mantener.
- Una vez tengamos los atributos con los que se ejecutará el algoritmo para la muestra *i-ésima*, se ordenan los índices con el propósito de conseguir una mayor claridad a la hora de mostrar por pantalla los resultados, y se guarda en el ArrayList *localAttIndexSubSpaceVectorBySample*.

Con estos pasos, se generará un subespacio para cada muestra que se haya generado.

```
for (int iSample = 0; iSample < samplesVector.length; iSample++) {  
  
    // Copy of the vector of indices in order to get the original instances back so  
    // we do the Shuffle again  
    indices = attIndexVector.clone();  
  
    // Shuffle function to randomly mix and move attributes  
    // (based on weka.classifiers.meta.RandomSubSpace.randomSubSpace() function  
    Collections.shuffle(Arrays.asList(indices), random);  
  
    // Temporal vector to save the attributes to keep for each sample  
    // We are left only with the "subSpaceSize" of the indexes, the total number of  
    // attributes to keep  
    Integer[] attIndexSubSpace = Arrays.copyOfRange(indices, 0, subSpaceSize);  
    // Sort the indices for the purpose of clarity in the classifier output  
    Arrays.sort(attIndexSubSpace);  
  
    // Save in the ArrayList the array of the attributes of the i-th sample to keep  
    localAttIndexSubSpaceVectorBySample.add(attIndexSubSpace);  
}
```

Figura 45: Cuarto segmento método buildClassifier - PCTRSM

En esta última figura 46, se continúa el algoritmo como el *PCTBagging*, el *J48PartiallyConsolidated*, creando primero los modelos del árbol base y consolidado. Después se llama al método *buildClassifier*, de la clase que se explicará posteriormente *C45RSMPartiallyConsolidatedPruneableClassifierTree*, limpiando luego el árbol consolidado y los árboles base consolidados.

```

/** Set the model selection method to determine the consolidated decisions */
ModelSelection modSelection;
// TODO Implement the option binarySplits of J48
modSelection = new C45RSMConsolidatedModelSelection(m_minNumObj, instances, m_useMDLcorrection,
    m_doNotMakeSplitPointActualValue);
/**
 * Set the model selection method to force the consolidated decision in each
 * base tree
 */
C45ModelSelectionExtended baseModelToForceDecision = new C45ModelSelectionExtended(m_minNumObj, instances,
    m_useMDLcorrection, m_doNotMakeSplitPointActualValue);
// TODO Implement the option reducedErrorPruning of J48
C45RSMPartiallyConsolidatedPruneableClassifierTree localClassifier =
    new C45RSMPartiallyConsolidatedPruneableClassifierTree(
        modSelection, baseModelToForceDecision, !m_unpruned, m_CF, m_subtreeRaising, !m_noCleanup,
        m_collapseTree, samplesVector.length, localAttIndexSubSpaceVectorBySample);

localClassifier.buildClassifier(instances, samplesVector, m_PCTBconsolidationPercent);

m_root = localClassifier;
m_Classifiers = localClassifier.getSampleTreeVector();
// // We could get any base tree of the vector as root and use it in the graphical interface
// // (for example, to visualize it)
// //m_root = localClassifier.getSampleTreeIth(0);

((C45ModelSelection) modSelection).cleanup();
((C45ModelSelection) baseModelToForceDecision).cleanup();

```

Figura 46: Quinto segmento método *buildClassifier* - PCTRSM

## Métodos secundarios

Como se ha mencionado en los métodos principales del clasificador, se ha necesitado crear dos funciones a parte de las pedidas por Weka para desarrollar el código. Estas dos funciones son las siguientes:

### numberOfAttributes

Esta función se encarga de calcular el tamaño del subespacio que utilizará el clasificador. Como se ve en la figura 47, la función recibe los valores de entrada, un *integer total*, con el número de atributos del espacio original, y un *double fraction*, que puede tomar los valores que se han explicado anteriormente para *m\_PCTRSMsubSpaceSize*. 0, para utilizar la función  $\text{int}(\log_2(\text{total}) + 1)$ , un valor  $<1$ , para calcular el porcentaje de los atributos y un valor  $>1$ , para indicar el valor absoluto de atributos a mantener.

```

protected int numberOfAttributes(int total, double fraction) {
    int k = 0;
    if(Utils.eq(fraction, (double)0.0)) { // If fraction == 0, int(log2(#predictors) + 1) is used.
        k = (int) Utils.log2(total) + 1;

    } else { // If fraction < 1, percentage. If fraction > 1, absolute number of attributes.
        k = (int) Math.round((fraction < 1.0) ? total*fraction : fraction);

        if (k > total)
            k = total;
        if (k < 1)
            k = 1;
    }
    return k;
}

```

Figura 47: Método numberOfAttributes - PCTRSM

## generateRSMSamples

El parámetro `m_PCTRSMsamplesGeneration` podía tomar dos valores al ser un *boolean*, *True* o *False*. En caso de elegir *False*, para generar el vector de submuestras se utilizará esta función. Este método se encarga de que se ejecute el algoritmo sin generación de muestras. Según el valor que se haya indicado en el parámetro `RMnumberSamples`, se realizarán tantas copias de los datos originales sin remuestreo, y se devolverá el vector de submuestras a utilizar.

```

protected Instances[] generateRSMSamples(Instances instances) throws Exception {

    int numberSamples = (int) getRMnumberSamples();
    Instances[] samplesVector = new Instances[numberSamples];

    // can classifier tree handle the data?
    getCapabilities().testWithFail(instances);

    // remove instances with missing class
    InstancesConsolidated instancesWMC = new InstancesConsolidated(instances);
    instancesWMC.deleteWithMissingClass();

    if (m_Debug) {
        System.out.println("=== Generation of the RSM set of samples ===");
    }
    /** Original sample size */
    int dataSize = instancesWMC.numInstances();
    if (dataSize == 0)
        System.err.println("Original data size is 0! Handle zero training instances!");
    else if (m_Debug)
        System.out.println("Original data size: " + dataSize);

    // Loop to save the original instances "numberSamples" times without
    // modifications
    for (int iNumberSample = 0; iNumberSample < numberSamples; iNumberSample++) {
        samplesVector[iNumberSample] = instances;
    }
    return samplesVector;
}

```

Figura 48: Método generateRSMSamples - PCTRSM



## ● Clases Secundarias

Para el correcto funcionamiento del PCTRSM, se ha necesitado crear 4 clases nuevas para poder aplicar las bases del *RSM*, los subespacios generados, a la hora de crear los árboles para cada submuestra. Estas 4 clases estarán incluidas en el paquete *Classifiers.Trees.PCTRSM*. Todas ellas han sido creadas para modificar métodos del J48, J48Consolidated o J48PartiallyConsolidated, que ahora necesitaban como parámetro el subespacio generado para cada muestra. Las clases son las siguientes:

- **Clase C45RSMConsolidatedModelSelection:** Esta es la clase para seleccionar una división de tipo *C4.5Consolidated* para un conjunto de datos dado, que extiende la clase *C45ConsolidatedModelSelection*. Es la encargada de la creación del árbol consolidado.
- **Clase C45RSMPartiallyConsolidatedPruneableClassifierTree:** Esta será la clase para manejar una estructura de árbol consolidada que se puede podar usando los procedimientos del C4.5, que extiende la clase *C45PartiallyConsolidatedPruneableClassifierTree*.
- **Clase C45RSMPruneableClassifierTreeExtended:** Clase para manejo extendido de la clase *C45PruneableClassifierTree*, que extiende la clase *C45PruneableClassifierTreeExtended*.

Estas tres clases casi no han sufrido modificaciones. Se han añadido los valores de los subespacios como parámetros para las funciones necesarias, para así generar el árbol consolidado en base a ellos. Ahora cada árbol votará la variable por la que dividiría el árbol, teniendo en cuenta que esa variable debe estar incluida en el subespacio.

Lo mismo ocurre para el árbol parcialmente consolidado. Una vez podado el árbol consolidado hasta el porcentaje fijado, se continuará con la generación del árbol con variables que deben estar incluidas en el subespacio.

Puede darse el caso en el que aparezca una variable que no se incluye en el subespacio de esa muestra en su árbol generado. Esto se debe a la parte consolidada, ya que la variable más votada para ese nivel ha sido una que está fuera de su rango de atributos. A partir de la parte consolidada de los árboles, todas variable estarán incluidas en su respectivo subespacio asignado.

- **Clase C45ToRSMModelSelection:** Esta es la clase en la que el subespacio entra en juego, que extiende la clase *C45ModelSelection*. En la función *selectModel*, se encarga de seleccionar la “mejor” división de los datos posible, buscando la “mejor” variable por la que dividir, en función a conceptos como la entropía, la ganancia de información... En este caso, se deberán realizar todos estos cálculos siempre y cuando la variable esté dentro del subespacio, si no, se continuará con el siguiente atributo directamente, como se ve en la figura 49.

```

// For each attribute.
for (i = 0; i < data.numAttributes(); i++) {

    // Apart from class attribute and is contained in the list of attributes of the sample to be kept
    if ((i != (data).classIndex()) &&
        (attIndexList.contains(i)) ) {

        // Get models for current attribute.
        currentModel[i] = new C45Split(i, m_minNoObj, sumOfWeights,
            m_useMDLcorrection);
        currentModel[i].buildClassifier(data);

        // Check if useful split for current attribute
        // exists and check for enumerated attributes with
        // a lot of values.
        if (currentModel[i].checkModel()) {
            if (m_allData != null) {
                if ((data.attribute(i).isNumeric()
                    || (multiVal || Utils.sm(data.attribute(i).numValues(),
                        (0.3 * m_allData.numInstances())))) {
                    averageInfoGain = averageInfoGain + currentModel[i].infoGain();
                    validModels++;
                }
            } else {
                averageInfoGain = averageInfoGain + currentModel[i].infoGain();
                validModels++;
            }
        }
    } else {
        currentModel[i] = null;
    }
}
}

```

Figura 49: Clase C45ToRSMMModelSelection - PCTRS

## 8.2 PCTRF

En este apartado se van a explicar las clases, parámetros y métodos que han sido necesarios para diseñar e implementar el código del PCTRF en WEKA.

Para el correcto funcionamiento del algoritmo, se ha necesitado crear una clase principal o base que se llama PCTRF, como el nombre del algoritmo, y 4 clases secundarias nuevas para poder aplicar las bases del *RF*, los subespacios generados a la hora de crear los árboles para cada nodo. Estas 4 clases estarán incluidas en el paquete *Classifiers.Trees.PCTRF*.

## • Clase Base: PCTRF

El nombre que se le ha dado a la clase principal ha sido PCTRF, el mismo nombre que al algoritmo. Se ha escogido como superclase el *J48PartiallyConsolidated* y se han añadido los dos parámetros nuevos que serán necesarios para el desarrollo del código. Esta clase pertenece al paquete de *classifiers.trees* de weka, al igual que el *J48*, el *J48Consolidated*, el *J48PartiallyConsolidated* y el *PCTRFM*.

```
package weka.classifiers.trees;

import java.util.Enumeration;

public class PCTRF
    extends J48PartiallyConsolidated
    implements OptionHandler, Drawable, Matchable, Sourcable,
               WeightedInstancesHandler, Summarizable, AdditionalMeasureProducer,
               TechnicalInformationHandler {

    /** for serialization */
    private static final long serialVersionUID = -8994701662253121033L;

    /** Sets the number of randomly chosen attributes. If 0,  $\text{int}(\log_2(\#\text{predictors}) + 1)$  is used.
     * Based on weka.classifiers.meta.RandomSubSpace.numberOfAttributes(int, double),
     * if less than 1 as a percentage of the number of attributes, otherwise the absolute number of attributes */
    protected float m_PCTRFnumFeatures = (float)0.5;

    /** Boolean to choose sample generation or not*/
    protected boolean m_PCTRFsamplesGeneration = true;
}
```

Figura 50: Definición de la clase y sus atributos - PCTRF

Como se puede ver en la figura 50, los dos parámetros añadidos han sido los siguientes:

- **m\_PCTRFnumFeatures:** Funciona de la misma manera que el parámetro *m\_PCTRFsubSpaceSize*. Este valor determina el número de atributos que se mantendrán para cada submuestra, como combinación de la implementación en WEKA de los algoritmos RSM y RF. Tiene 3 tipos de valores distintos que puede aceptar:
  - **Valor 0:** Utilizará la función  $\text{int}(\log_2(\#\text{predictors}) + 1)$ , siendo *predictors* el número de atributos de los datos.
  - **Valor <1:** En caso de utilizar un valor decimal, un valor menor que 1 y mayor que 0, se interpretará como el porcentaje de atributos del espacio original a mantener, pero en tanto por 1. Por defecto *m\_PCTRFnumFeatures* tiene un valor de 0.5, igual que el algoritmo RF, que en WEKA es el Random Forest (su implementación), para indicar que se seleccionará el 50% de los atributos existentes.

- **Valor >1:** Si el valor que se utiliza es mayor que 1, será el total o valor absoluto de atributos a mantener, sin importar el número de atributos originales. Si se indica un valor mayor al total de atributos, se mantendrá el total de los mismos.

Los métodos utilizados para implementar este parámetro son los siguientes, como se puede ver en las figuras 51, 52 y 53:

```
/**
 * Returns the tip text for this property
 * @return tip text for this property suitable for
 * displaying in the explorer/experimenter gui
 */
public String PCTRFnumFeaturesTipText() {
    return "Sets the number of randomly chosen attributes.\n"
        + "If 0, int(log_2(#predictors) + 1) is used.\n"
        + "If less than 1 as a percentage of the number of attributes,\n"
        + "otherwise the absolute number of attributes.";
}
```

Figura 51: Método PCTRFnumFeaturesTipText

```
/**
 * Get the value of numFeatures.
 *
 * @return Value of numFeatures.
 */
public float getPCTRFnumFeatures() {
    return m_PCTRFnumFeatures;
}
```

Figura 52: Método getPCTRFnumFeatures

```
/**
 * Set the value of numFeatures.
 *
 * @param v Value to assign to numFeatures.
 */
public void setPCTRFnumFeatures(float v) throws Exception {
    m_PCTRFnumFeatures = v;
}
```

Figura 53: Método setPCTRFnumFeatures

- **m\_PCTRFsamplesGeneration:** Es el mismo parámetro que para el algoritmo PCTRS. Con este valor se determinará si va a realizar una generación de muestras heredada del J48Consolidated o se mantiene el estado original de los datos como en el RandomSubSpace.
  - **True:** Valor por defecto del clasificador. Si se elige este método, se realizará una generación de muestras.
  - **False:** En caso de no elegir una generación de muestras, la base de datos original no será modificada y todas las submuestras serán idénticas.

Los métodos utilizados para implementar este parámetro son los siguientes, como se puede observar en las figuras 54, 55 y 56:

```
/**
 * Returns the tip text for this property
 * @return tip text for this property suitable for
 * displaying in the explorer/experimenter gui
 */
public String PCTRFsamplesGenerationTipText() {
    return "Samples generation mode. true for j48PartiallyConsolidated samples generation type. "
        + "If false is selected, m_PCTRFnumberSamples"
        + "will be the number of samples to generate ";
}
```

Figura 54: Método PCTRFsampleGenerationTipText

```
/**
 * Get the value of samplesGeneration.
 *
 * @return Value of samplesGeneration.
 */
public boolean getPCTRFsamplesGeneration() {
    return m_PCTRFsamplesGeneration;
}
```

Figura 55: Método getPCTRFsamplesGeneration

```
/**
 * Set the value of samplesGeneration.
 *
 * @param samplesGeneration Value to assign to samplesGeneration.
 */
public void setPCTRFsamplesGeneration(boolean samplesGeneration) throws Exception {
    if (!samplesGeneration) {
        setRMnumberSamplesHowToSet(new SelectedTag(NumberSamples_FixedValue, TAGS_WAYS_TO_SET_NUMBER_SAMPLES));
        RandomForest rf = new RandomForest();
        setRMnumberSamples((float) rf.getNumIterations());
    }
    /**
     * If samplesGeneration is true the user can choose to set the number of samples
     * by 'using a fixed value' or 'based on a coverage value (%)'.
     */
    m_PCTRFsamplesGeneration = samplesGeneration;
}
```

Figura 56: Método setPCTRFsamplesGeneration

## Métodos principales

Al igual que para *PCTRF*, para que Weka detecte y ejecute nuestros algoritmos, es necesaria la implementación de varios métodos obligatorios para que así pueda reconocer e incluir el clasificador en las diferentes aplicaciones. También habrá algunos de estos métodos que no se hayan implementado, ya que siguen el mismo proceso que los métodos que ha heredado.

### listOptions

Método para devolver las opciones del clasificador, junto a las opciones de las clases que extiende.

```
public Enumeration<Option> listOptions() {  
  
    Vector<Option> newVector = new Vector<Option>();  
  
    // J48, J48Consolidated and J48PartiallyConsolidated options  
    // =====  
    Enumeration<Option> en;  
    en = super.listOptions();  
    while (en.hasMoreElements())  
        newVector.addElement((Option) en.nextElement());  
  
    // PCTRF options  
    // =====  
    newVector.  
    addElement(new Option(  
        "\tNumber of features to keep:\n" + "\t(default 0.5)"  
        + "\t\t< 1: percentage of the number of attributes\n"  
        + "\t\t>=1: absolute number of attributes\n"  
        + "\t\t==0: int(log_2(#predictors) + 1)\n",  
        "PCTRF-K", 1, "-PCTRF-K <number of features>"));  
  
    newVector.  
    addElement(new Option(  
        "\tSamples generation mode\n" +  
        "\t(default true)",  
        "PCTRF-S", 1, "-PCTRF-S <Samples generation>"));  
  
    return newVector.elements();  
}
```

Figura 57: Método listOptions - PCTRF

## setOptions

Esta es la función que se encarga de guardar los valores de las opciones o parámetros del clasificador y tratar las excepciones en relación a ellas.

```
public void setOptions(String[] options) throws Exception {

    String PCTRFnumFeatures = Utils.getOption("PCTRF-K", options);
    if(PCTRFnumFeatures.length()!=0) {
        setPCTRFnumFeatures(Float.valueOf(PCTRFnumFeatures));
    }else {
        setPCTRFnumFeatures((float)0.5);
    }

    String PCTRFsamplesGeneration = Utils.getOption("PCTRF-S", options);
    if(PCTRFsamplesGeneration.length()!=0) {
        setPCTRFsamplesGeneration(Boolean.valueOf(PCTRFsamplesGeneration));
    }else {
        setPCTRFsamplesGeneration(true);
    }

    // J48, J48Consolidated and J48PartiallyConsolidated options
    // =====
    super.setOptions(options);
}
```

Figura 58: Método setOptions - PCTRF

## getOptions

Esta función devuelve un array de strings con las opciones o la configuración que compone el clasificador (según el formato de la línea de comandos). Las opciones deben coincidir con las fijadas en el método setOptions.

```
public String [] getOptions() {

    Vector<String> result = new Vector<String>();

    // J48, J48Consolidated, J48PartiallyConsolidated options
    // =====
    String[] options = super.getOptions();
    for (int i = 0; i < options.length; i++)
        result.add(options[i]);

    // PCTRF Options
    // =====
    result.add("-PCTRF-K");
    result.add(" " + m_PCTRFnumFeatures);

    result.add("-PCTRF-S");
    result.add(" " + m_PCTRFsamplesGeneration);

    return (String[]) result.toArray(new String[result.size()]);
}
```

Figura 59: Método getOptions - PCTRF

## globalInfo

Con el método global info se devuelve una pequeña descripción general del algoritmo.

```
public String globalInfo() {
    return "Class for generating a pruned or unpruned J48PartiallyConsolidated "
        + "decision tree with Random Forest"
    + ". For more information, see\n\n" + getTechnicalInformation().toString();
}
```

Figura 60: Método globalInfo - PCTRF

## buildClassifier

Este método es la función principal de esta clase. Es el encargado de generar el modelo de nuestro clasificador.

Como se ve en la figura 61, comienza comprobando si el algoritmo es capaz de procesar los datos. Si esto es posible, eliminará los casos sin clase definida y continuará creando el vector de submuestras.

```
public void buildClassifier(Instances instances)
    throws Exception {

    // can classifier tree handle the data?
    getCapabilities().testWithFail(instances);

    // remove instances with missing class before generate samples
    instances = new Instances(instances);
    instances.deleteWithMissingClass();
}
```

Figura 61: Primer segmento método buildClassifier - PCTRF

En función a la opción que se haya elegido para el parámetro m\_PCTRFsamplesGeneration, ver figura 62, se realizará una generación de muestras, al igual que se realiza en el algoritmo J48Consolidated, o se llamará a la función generatePCTRFsamples, que se explicará en el siguiente apartado.



```

//Generate as many samples as the number of samples with the given instances
Instances[] samplesVector;

if (m_PCTRFsamplesGeneration) { // if PCTRFsamplesGeneration is true, regular samples generation of
    // j48Consolidated
    samplesVector = generateSamples(instances);
} else { // if PCTRFsamplesGeneration is false, no sampling generation
    samplesVector = generateRFSamples(instances);
}

```

Figura 62: Segundo segmento método buildClassifier - PCTRF

Como se puede observar en la figura 63, con el vector de submuestras ya creado, se necesita obtener una serie de valores para poder generar los árboles posteriores siguiendo el método del Random Forest. Los cuatro valores que se necesitan son:

- *Array* con los índices del espacio original de atributos: *attIndexVector*
- El número de atributos a mantener: *numFeatures*. Este dato se obtiene mediante la función `numberOfAttributes`.
- Un objeto *Random* *random* generado a partir de la *seed* fijada.
- Un vector de objetos *Random*, *randomVector*, con longitud igual al número de submuestras que se van a generar.

```

// Array of the attributes indices// {0,1,2,3,4...}
// (based on weka.classifiers.meta.RandomSubSpace.buildClassifier(Instances))
Integer[] attIndexVector = new Integer[instances.numAttributes() - 1];
int classIndex = instances.classIndex();
int offset = 0;
for (int i = 0; i < attIndexVector.length; i++) {
    if (i != classIndex) {
        attIndexVector[offset++] = i;
    }
}

//Number of attributes to keep
int numFeatures = numberOfAttributes(attIndexVector.length, m_PCTRFnumFeatures);

// Generate a random object with the seed
Random random = instances.getRandomNumberGenerator(m_Seed);

// Vector of random objects for each sample
Random[] randomVector = new Random[samplesVector.length];

```

Figura 63: Tercer segmento método buildClassifier - PCTRF

Cuando todas las variables necesarias estén definidas, con un bucle que irá desde 0 hasta la longitud del vector de muestras, se asignará un objeto *Random* a cada una de las submuestras, ver figura 64. Para ello, con el primer objeto random creado a partir de la seed, se creará por cada iteración un nuevo objeto random haciendo uso de la función *.nextInt()* y se almacenará en el vector *randomVector*.

```
for (int iSample = 0; iSample < samplesVector.length; iSample++) {
    randomVector[iSample] = new Random(random.nextInt());
}
```

Figura 64: Cuarto segmento método buildClassifier - PCTRF

En esta última figura 65, se continúa el algoritmo como el *J48PartiallyConsolidated*, el *PCTBagging*, creando primero los modelos del árbol base y consolidado. Después se llama al método *buildClassifier*, de la clase que se explicará posteriormente *C45RFPartiallyConsolidatedPruneableClassifierTree*, limpiando luego el árbol consolidado y los árboles base consolidados.

```
/** Set the model selection method to determine the consolidated decisions */
ModelSelection modSelection;
// TODO Implement the option binarySplits of J48
modSelection = new C45RFConsolidatedModelSelection(m_minNumObj, instances,
    m_useMDLcorrection, m_doNotMakeSplitPointActualValue);
/** Set the model selection method to force the consolidated decision in each base tree*/
C45ModelSelectionExtended baseModelToForceDecision = new C45ModelSelectionExtended(m_minNumObj, instances,
    m_useMDLcorrection, m_doNotMakeSplitPointActualValue);
// TODO Implement the option reducedErrorPruning of J48
C45RFPartiallyConsolidatedPruneableClassifierTree localClassifier =
    new C45RFPartiallyConsolidatedPruneableClassifierTree(modSelection, baseModelToForceDecision,
        !m_unpruned, m_CF, m_subtreeRaising, !m_noCleanup, m_collapseTree, samplesVector.length,
        randomVector, attIndexVector, numFeatures);

localClassifier.buildClassifier(instances, samplesVector, m_PCTBconsolidationPercent);

m_root = localClassifier;
m_Classifiers = localClassifier.getSampleTreeVector();
// // We could get any base tree of the vector as root and use it in the graphical interface
// // (for example, to visualize it)
// //m_root = localClassifier.getSampleTreeIth(0);

((C45ModelSelection) modSelection).cleanup();
((C45ModelSelection) baseModelToForceDecision).cleanup();
```

Figura 65: Quinto segmento método buildClassifier - PCTRF

## ● Clases Secundarias

Para el correcto funcionamiento del PCTRF, se ha necesitado crear 4 clases nuevas para poder aplicar las bases del *RF* a la hora de crear los árboles para cada submuestra. Estas 4 clases estarán incluidas en el paquete *Classifiers.Trees.PCTRF*. Todas ellas han sido creadas para modificar métodos del *J48Consolidated* y el *J48PartiallyConsolidated*, que ahora necesitaban como parámetro el array de objetos *Random*, el array de índices y el número de atributos a mantener para cada muestra. Las clases son las mismas que las creadas para el *PCTRSM*, pero ahora con la necesidad de más y diferentes parámetros:

- **Clase *C45RFConsolidatedModelSelection*:** Esta es la clase para seleccionar una división de tipo *C4.5Consolidated* para un conjunto de datos dado. Es la encargada de la creación del árbol consolidado.
- **Clase *C45RFPartiallyConsolidatedPruneableClassifierTree*:** Esta será la clase para manejar una estructura de árbol consolidada que se puede podar usando los procedimientos del *C4.5*.
- **Clase *C45RFPruneableClassifierTreeExtended*:** Clase para manejo extendido de la clase *C45PruneableClassifierTree*.

Estas tres clases casi no han sufrido modificaciones. Se han añadido los valores del array de objetos *Random*, el array de índices y el número de atributos a mantener como parámetros para las funciones necesarias, para así generar el árbol consolidado en base a ellos. Ahora cada árbol votará la variable por la que dividiría el árbol, teniendo en cuenta que esa variable debe estar incluida en el subespacio.

Lo mismo ocurre para el árbol parcialmente consolidado. Una vez podado el árbol consolidado hasta el porcentaje fijado, se continuará con la generación del árbol con variables que deben estar incluidas en el subespacio.

- **Clase *C45ToRFModelSelection*:** Esta es la clase en la que el subespacio se debe crear y entra en juego. En la función *selectModel*, se encarga de seleccionar la “mejor” división de los datos posible, buscando la “mejor” variable por la que dividir, en función a conceptos como la entropía, la ganancia de información... En este caso, se deberán realizar todos estos cálculos siempre y cuando la variable esté dentro del subespacio.

Tal como se ve en la figura 66, cada vez que se llame a esta función se realiza un shuffle sobre el vector de índices original con el objeto *random* que esté asignado a esa muestra, mezclando los valores de los índices de los atributos. Posteriormente, se tomarán solamente los “*numFeatures*” primeros valores, para así generar el subespacio para dicho nodo del árbol.

```

// Keep only the first "subSpaceSize" number of attributes after shuffling the indexes vector
Collections.shuffle(Arrays.asList(indexVector), irandom);

Integer[] attIndexSubSpace = Arrays.copyOfRange(indexVector, 0, numFeatures);

```

Figura 66: Primer segmento de C45ToRFModelSelection

Una vez generado el subespacio para el nodo, se obtiene la “mejor” variable por la que dividir el árbol, siempre y cuando se encuentre dentro de este subespacio, como se puede observar en la figura 67.

```

// For each attribute.
for (i = 0; i < data.numAttributes(); i++) {

    // Apart from class attribute.
    if ((i != (data).classIndex()) && (Arrays.asList(attIndexSubSpace).contains(i)) ) {

        // Get models for current attribute.
        currentModel[i] = new C45Split(i, m_minNoObj, sumOfWeights,
            m_useMDLcorrection);
        currentModel[i].buildClassifier(data);
    }
}

```

Figura 67: Primer segmento de C45ToRFModelSelection

## 9. EXPERIMENTACIÓN

Una vez implementados los algoritmos PCTRSM y PCTRF en WEKA, se ha experimentado con ellos para ver los resultados, así como su rendimiento y compararlos con sus competidores más cercanos.

Se ha hecho un *5-fold Cross-Validation* 10 veces para calcular las estimaciones de los criterios de bondad elegidos, además se han utilizado 33 bases de datos para la experimentación, que se divide en dos partes: Experimentación para elegir el mejor *Random Subspace Method (RSM)* y el mejor *RandomForest (RF)* para el *PCTRSM* y el *PCTRF*, respectivamente, y Experimentación para comparar el *PCTRSM* y el *PCTRF* con sus competidores más directos.

Al lado de todas las gráficas que se muestran habrá una leyenda indicando el algoritmo que hace referencia a cada línea, siendo los algoritmos en la parte los que mejores resultados han logrado obtener.

### 9.1 EXPERIMENTACIÓN PARA ELEGIR LOS MEJORES RSM Y RF

Como su nombre indica, este es el experimento que se ha realizado para encontrar las versiones de RSM y RF que mejor rendimiento tienen, y así, en la siguiente experimentación, utilizar PCTRSM y PCTRF con dichas versiones de RSM y RF, añadiendo además otros algoritmos base para realizar la comparación.

Los algoritmos son los siguientes:

- J48
- J48Consolidated con valores por defecto: Balanced coverage = 99 %
- J48Consolidated con 10, 30, 50 y 100 muestras Bootstrap
- Bagging con valores por defecto, REPTree como algoritmo base y 10 muestras Bootstrap
- Bagging con J48 como algoritmo base y 10, 30, 50 y 100 muestras Bootstrap
- RSM con valores por defecto, con REPTree como algoritmo base, 50% de atributos y 10 muestras sin remuestreo
- RSM con J48 como algoritmo base, con 10, 30, 50 y 100 muestras sin remuestreo y con un porcentaje de atributos de 10%, 25%, 50%, 75% y 90%.

- Random Forest con valores por defecto, con J48 como algoritmo base, con un porcentaje igual al  $\log_2(\text{total de atributos}) + 1$  y 100 muestras sin remuestreo
- Random Forest con 10, 30, 50, y 100 muestra sin remuestreos y con un porcentaje de atributos de 10%, 25%, 50%, 75%, y 90%.

Y estos son los criterios de bondad utilizados para comparar en este primer experimento:

- Tasa de acierto/Accuracy (Acc).
- Balanced Accuracy.
- Kappa.
- AUC (Area under ROC curve)

Para esta fase de la experimentación se han elegido medidas que solo miden la capacidad clasificatoria, ya que el objetivo de esta experimentación es saber qué algoritmo es el que “mejor” clasifica.

## Accuracy

La primera gráfica, ver figura 68, son los resultados del RF en cuanto al porcentaje de atributos junto al J48 como clasificador de referencia.

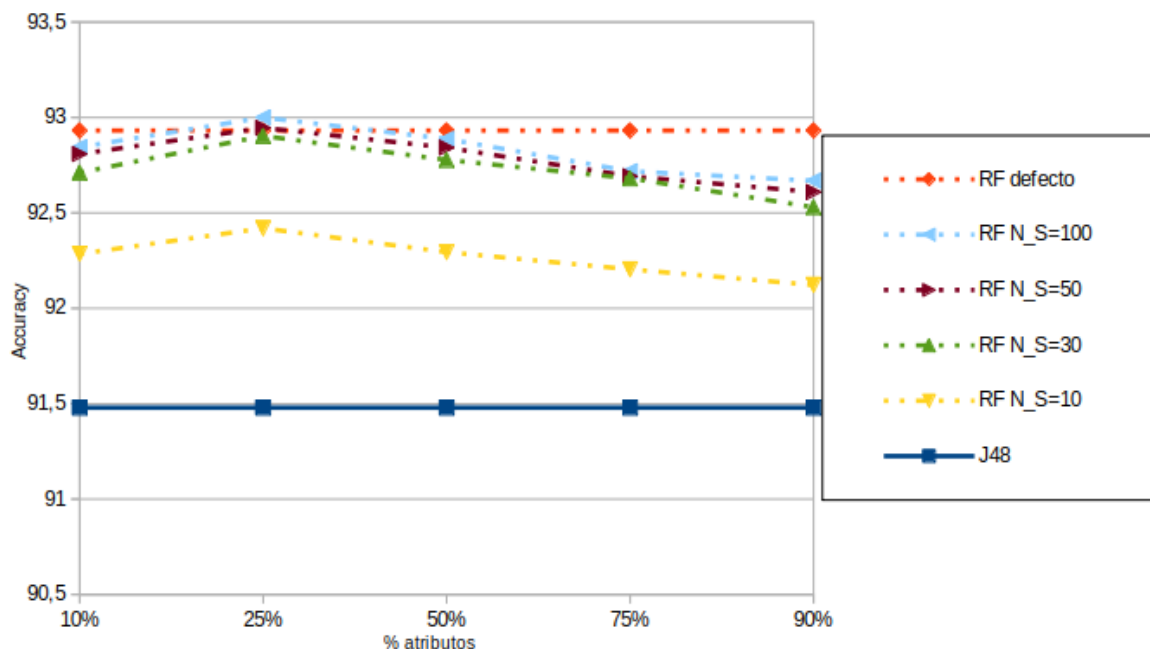


Figura 68: Gráfica Accuracy RF - Primera Fase

En la segunda gráfica, como se puede observar en la figura 69, aparecen los resultados de RSM junto a los de RF y el J48, también en función al porcentaje de los atributos.

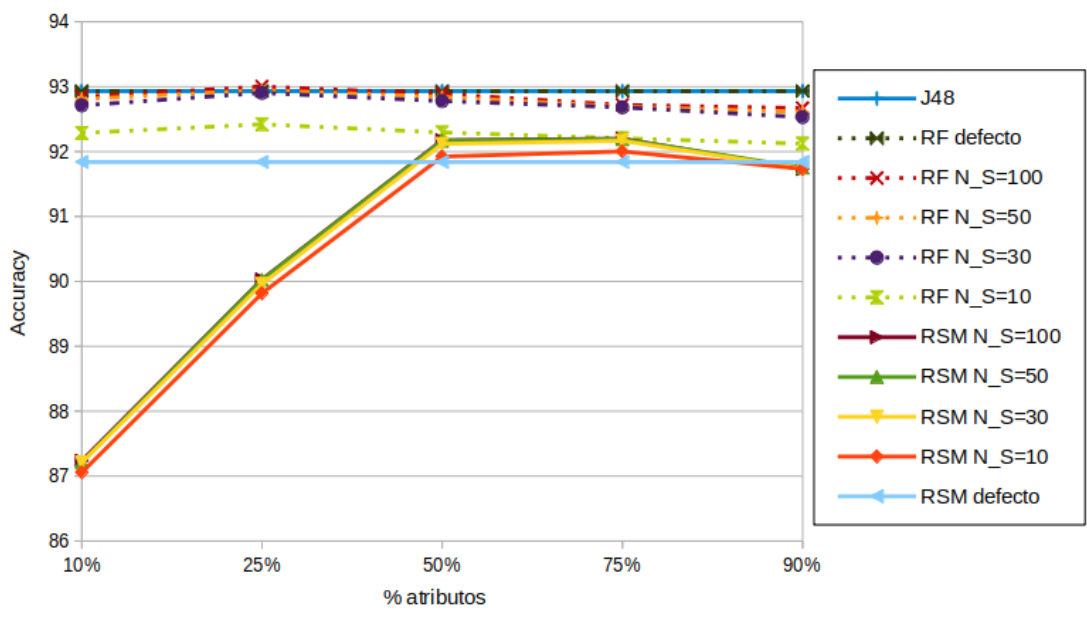


Figura 69: Gráfica Accuracy RSM y RF - Primera Fase

Para la medida Accuracy, los mejores resultados del RSM se obtienen con un porcentaje de atributos del 50-75%, y para RF con un porcentaje de atributos de 25-50%. En la figura 70 se muestran los resultados de estos valores junto a los del J48, J48Consolidated y Bagging, en función al número de muestras (N\_S) utilizadas en la construcción del clasificador.

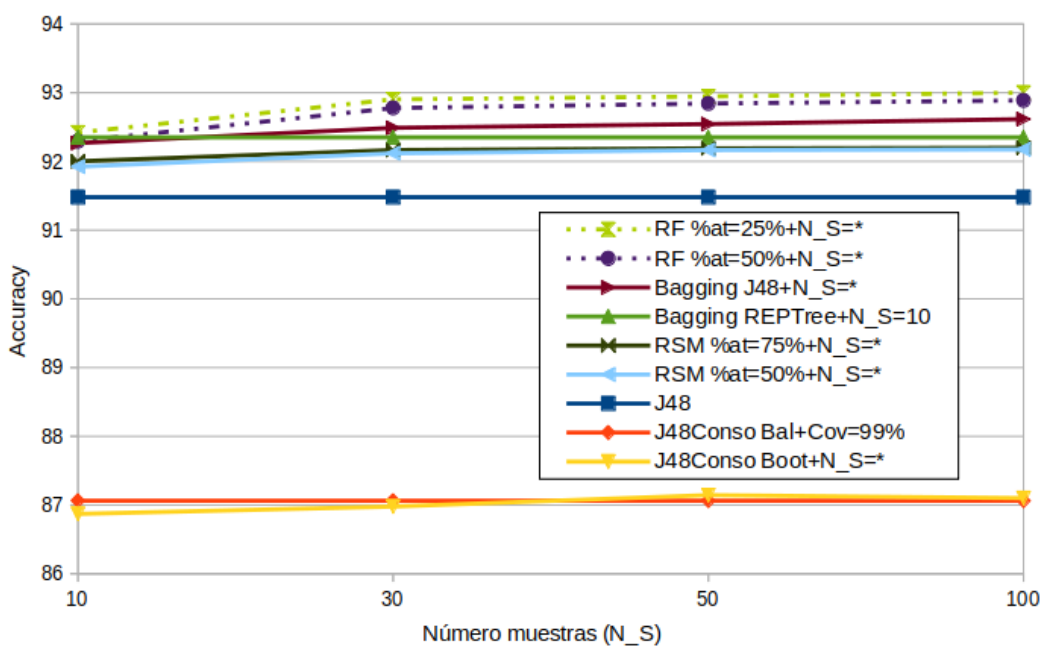


Figura 70: Gráfica Accuracy General - Primera Fase

En cuanto al Accuracy se refiere, se puede ver que los mejores resultados los obtiene el Random Forest, seguido del Bagging. El RSM, aunque se coloque por debajo, mantiene una diferencia mínima respecto al RF y Bagging, al igual que con el J48. A su vez, todos estos algoritmos obtienen buenos resultados alejándose del J48Consolidated con una mejoría mínima de un 4% para la Tasa de Acierto.

## Balanced Accuracy

En la figura 71 se puede observar la gráfica con los resultados del RF en función al porcentaje de los atributos usando el J48 como clasificador base.

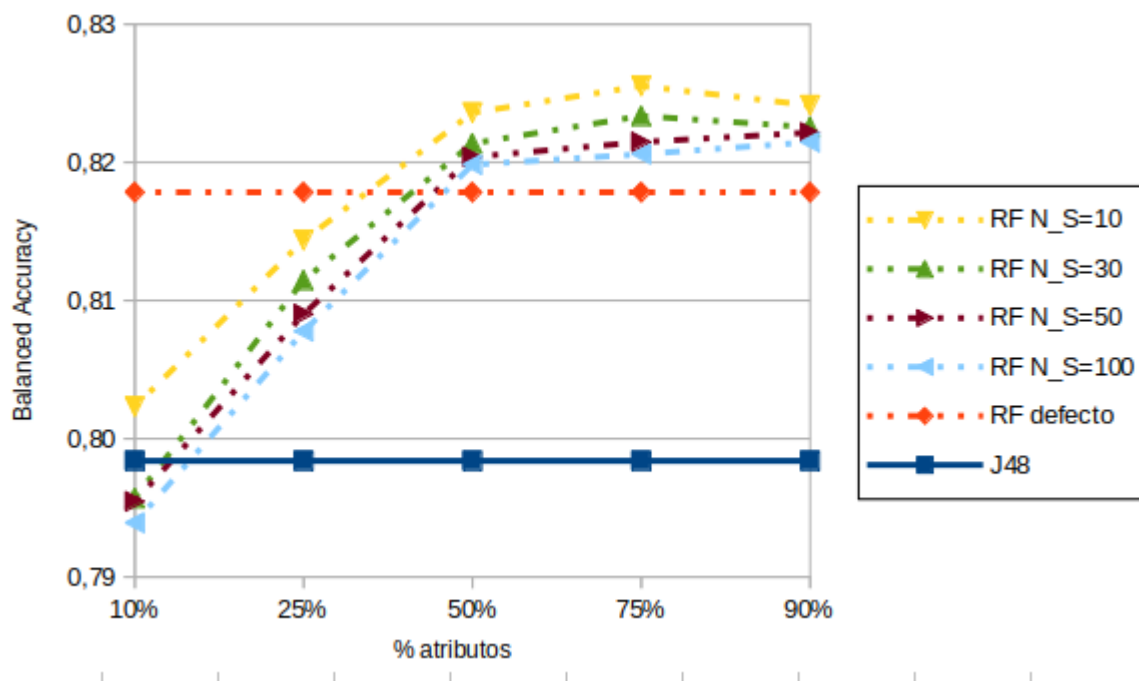


Figura 71: Gráfica Balanced Accuracy RF - Primera Fase

En la segunda gráfica, ver figura 72, se muestran los resultados del RSM respecto al porcentaje de atributos junto a los de RF y J48.



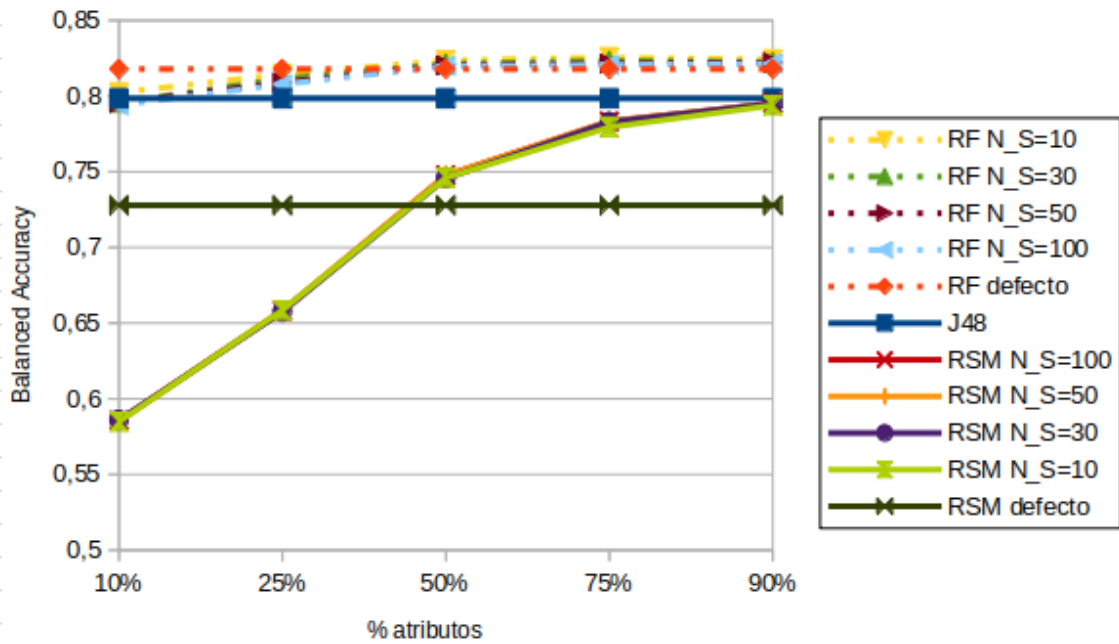


Figura 72: Gráfica Balanced Accuracy RSM y RF - Primera Fase

Para la medida Balanced Accuracy, los mejores resultados del RSM y RF se obtienen con un porcentaje de atributos del 75-90%. En la figura 73 se muestran los resultados de estos valores junto a los del J48, J48Consolidated y Bagging, en función al número de muestras utilizadas en la construcción del clasificador ( $N_S$ ).

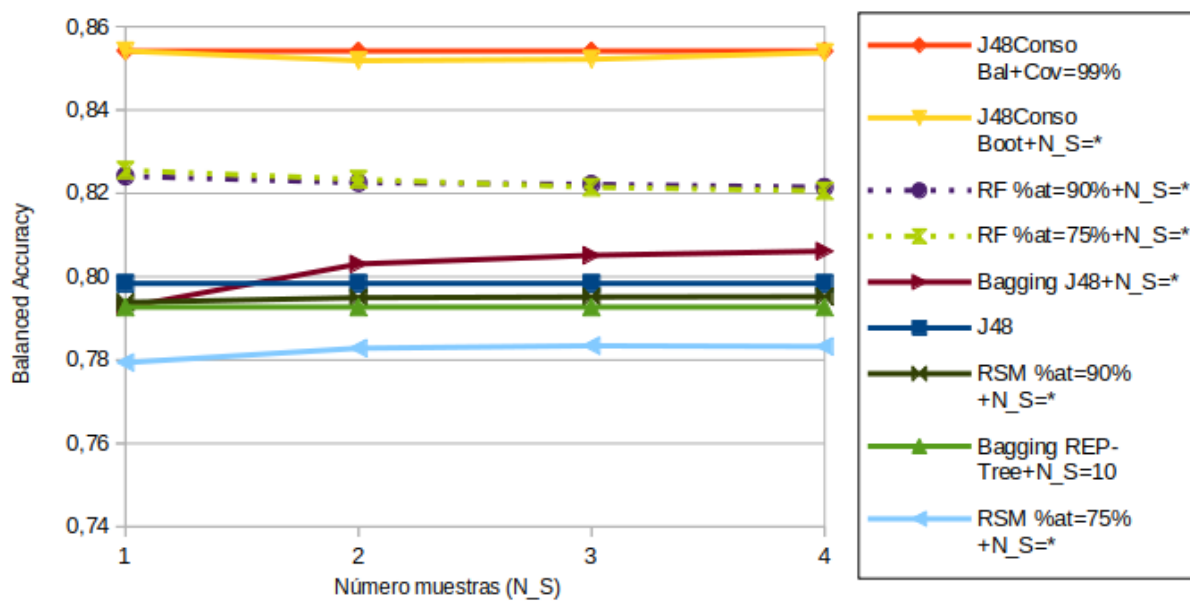


Figura 73: Gráfica Balanced Accuracy General - Primera Fase

En cuanto al Balanced Accuracy se refiere, se puede ver que los mejores resultados los obtiene el J48Consolidated, con muestras Balanceadas y un coverage del 99% y con muestras Bootstrap, seguido de ambos RF. Estos dos algoritmos obtienen una clara mejora sobre el resto, con el RSM con un 75% de atributos siendo el clasificador con peores resultados.

## AUC

Como se puede observar en la figura 74, la gráfica nos muestra los resultados del RF respecto al porcentaje de atributos junto al J48 como clasificador base.

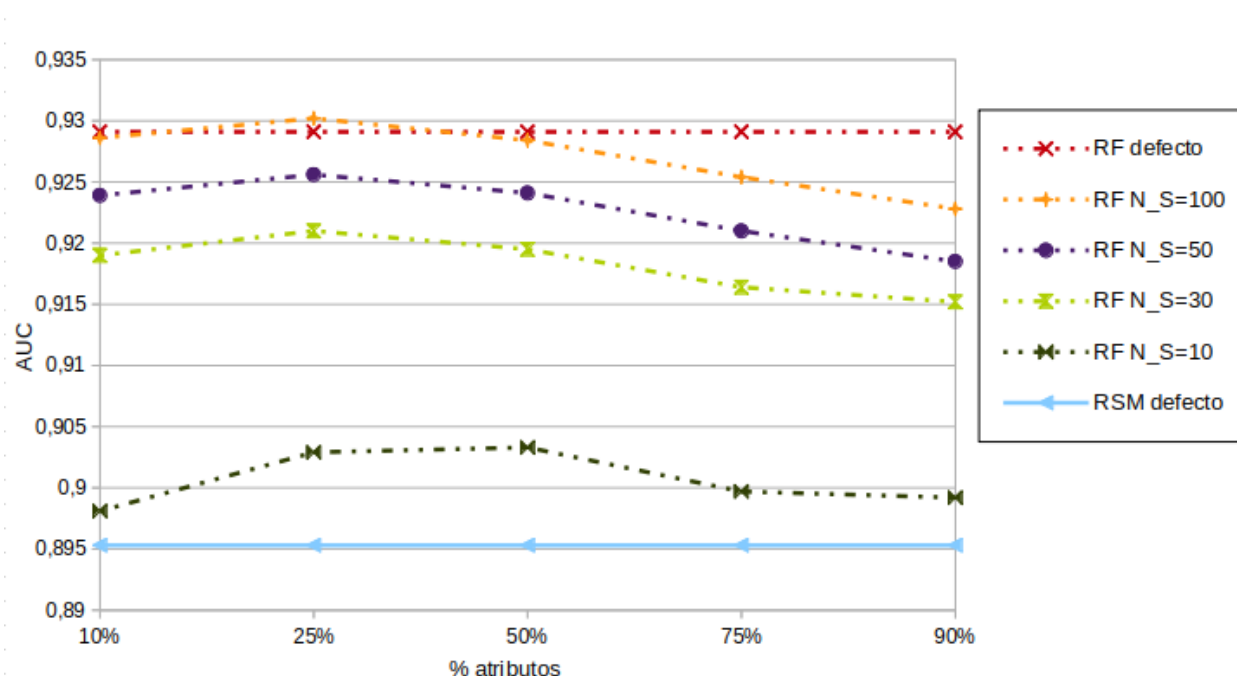


Figura 74: Gráfica AUC RF - Primera Fase

En la segunda gráfica, como se puede observar en la figura 75, aparecen los resultados RSM junto a los de RF y el J48.

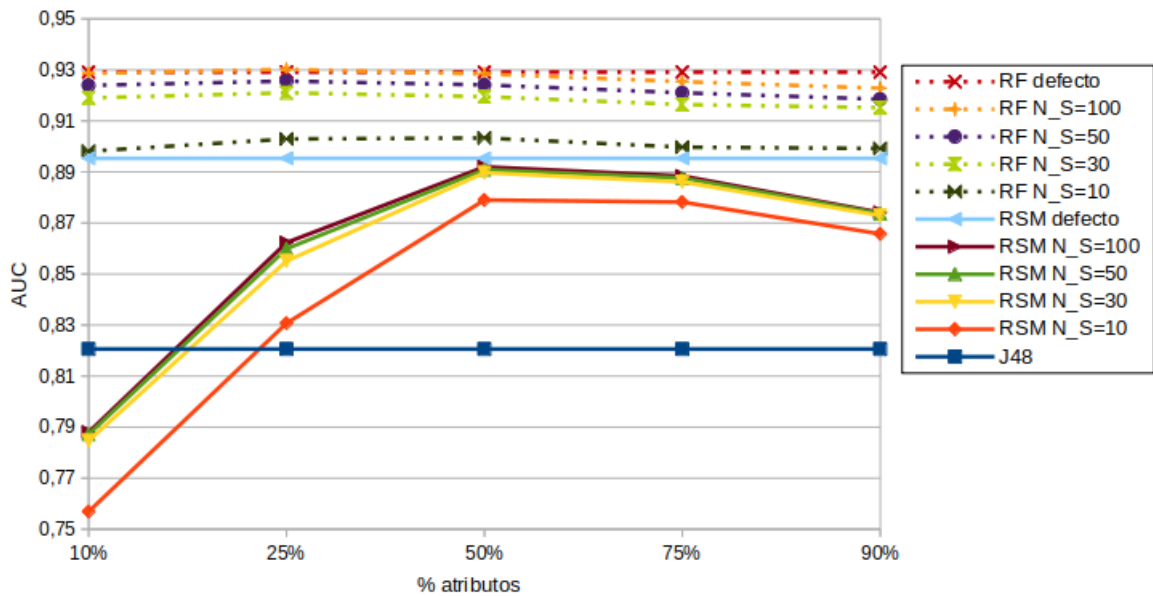


Figura 75: Gráfica AUC RSM y RF - Primera Fase

Para la medida AUC, los mejores resultados del RSM se obtienen con un porcentaje de atributos del 25-50%, y para RF con un porcentaje de atributos de 50-75%. En la figura 76 se muestran los resultados de estos valores junto a los del J48, J48Consolidated y Bagging, en función al número de muestras utilizadas.

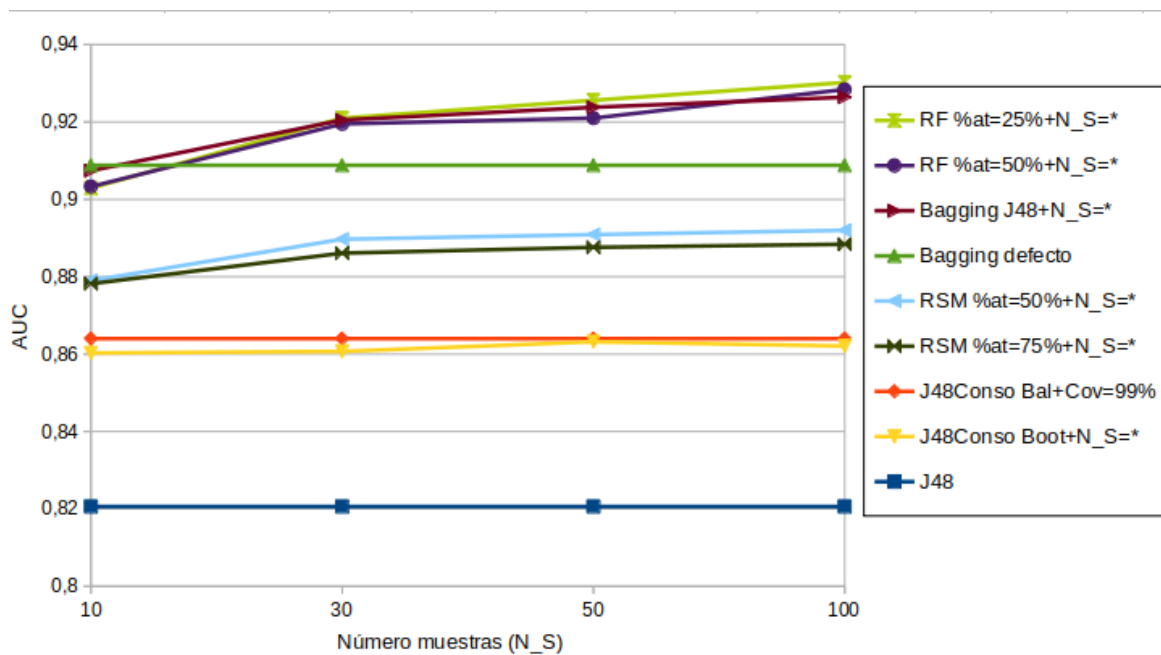


Figura 76: Gráfica AUC General - Primera Fase

Para la medida AUC, el RF y Bagging con J48 como clasificador base se colocan arriba con resultados muy similares, seguido del Bagging con valores por defecto. El RSM consigue mejorar al J48Consolidated, siendo el J48 el clasificador con peores resultados.

## KAPPA

La primera gráfica, ver figura 77, son los resultados en cuanto al porcentaje de atributos del RF con el J48 como clasificador de referencia.

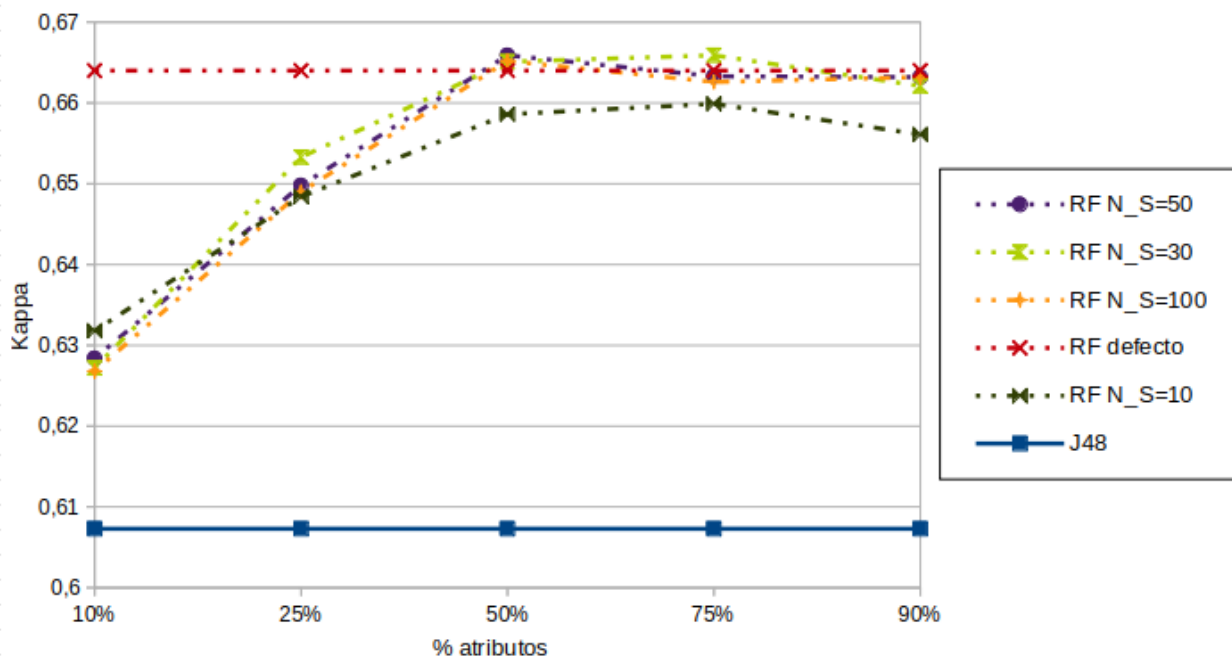


Figura 77: Gráfica Kappa RF - Primera Fase

En la figura 78 se puede ver la gráfica con los resultados del RSM en función al porcentaje de atributos junto a los de RF y J48.

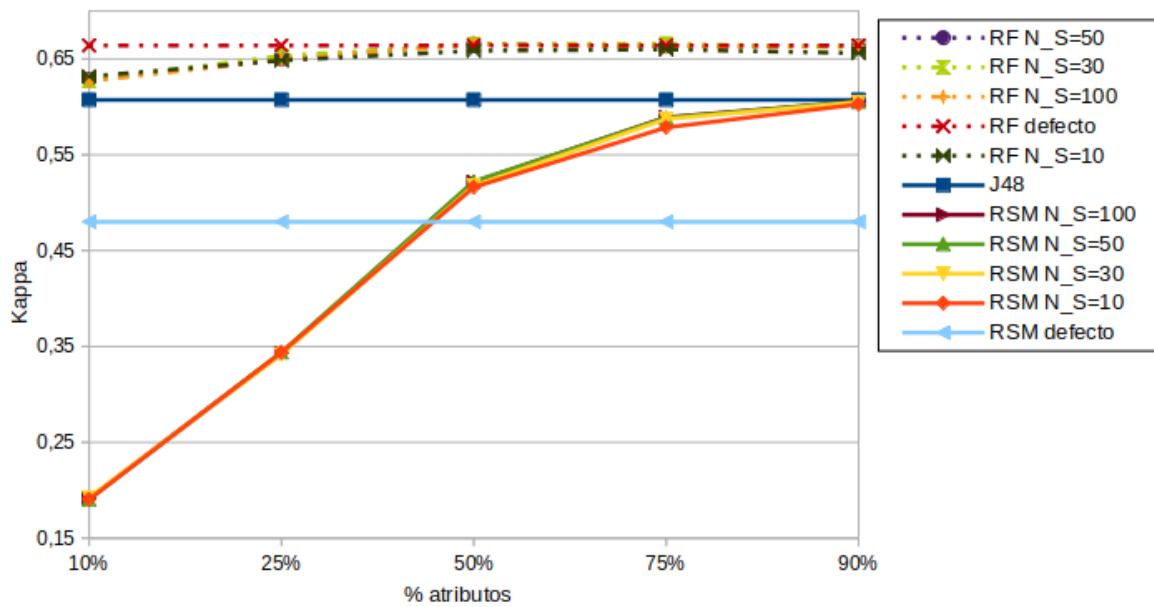


Figura 78: Gráfica Kappa RSM y RF - Primera Fase

Para la medida Kappa, los mejores resultados del RSM y RF se obtienen con un porcentaje de atributos del 50-75%. En la figura 79 se muestran los resultados de estos valores junto a los del J48, J48Consolidated y Bagging, en función al número de muestras generadas.

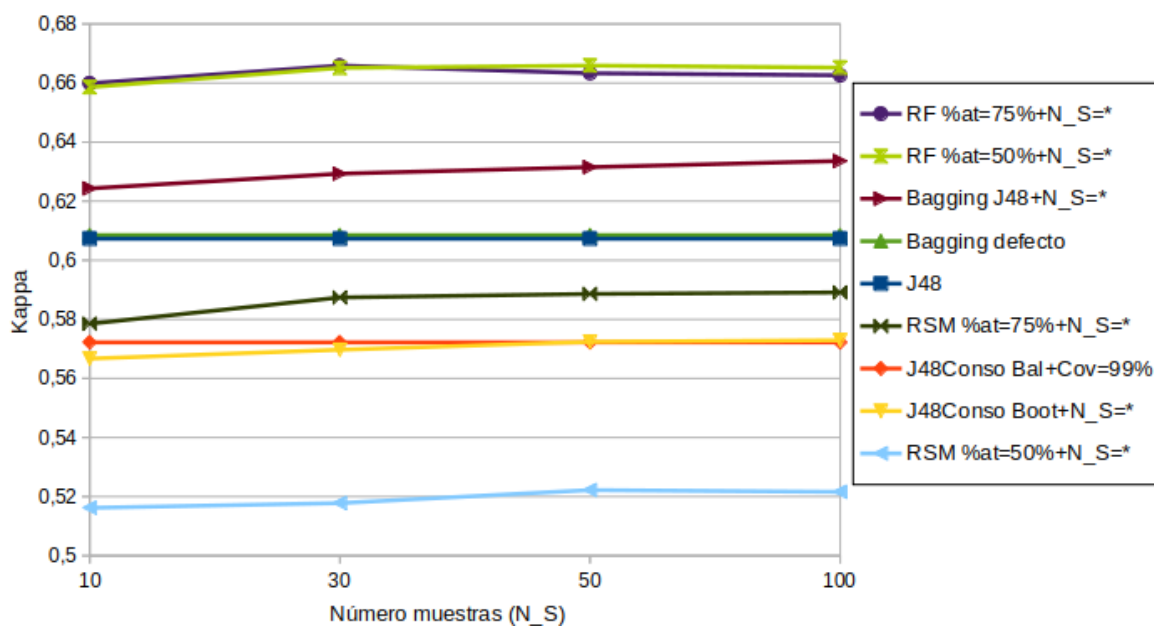


Figura 79: Gráfica Kappa General - Primera Fase

En cuanto a la medida Kappa se refiere, el RF con sus dos variaciones obtiene los mejores resultados. El Bagging, también con sus dos variaciones se sitúa seguido con el J48 debajo. También se observa que aunque el RSM con un 75% de atributos mejore ambos J48Consolidated, el RSM con un 50% se coloca bastante por debajo del resto.

### 9.1.1 CONCLUSIONES FASE 1

En todas las medidas se puede observar que los valores fluctúan en función al clasificador y los parámetros elegidos. Por lo general, el RF tiende a obtener buenos resultados en comparación al RSM, que suele situarse en la parte media-baja de las gráficas.

Uno de los factores a tener en cuenta, es el número de muestras a generar. Tanto para RF como para RSM, incluso para Bagging, hay una tendencia ascendente a medida que aumentan las bases de datos, siendo con un número de **100 muestras** el valor con el que mejores resultados se obtienen.

En las gráficas en las que se incluyen todas las variaciones de RF y RSM, se puede ver que aunque el porcentaje de atributos varía para obtener los mejores valores para cada medida, los dos porcentajes más recurrentes para ambos clasificadores son un **50%** y un **75%** de los atributos. Los valores del 10% y 25% son descartados rápidamente, ya que el subespacio que se genera parece ser demasiado pequeño y los resultados son peores en comparación a los otros, y el 90% parece no influir demasiado por lo general, siendo un subespacio demasiado grande.

## 9.2 EXPERIMENTACIÓN PARA COMPARAR PCTRSM Y PCTRF CON SUS COMPETIDORES MÁS DIRECTOS

Una vez realizado este primer experimento, ahora se va a comparar más detenidamente el PCTRSM y el PCTRF con sus mayores competidores: J48, J48Consolidated y Bagging, con los parámetros con los que mejores resultados se obtuvieron en la primera fase, pero sobre todo con el algoritmo en el que se ha basado, el PCTBagging.

Los algoritmos junto a sus parámetros son los siguientes:

- J48 - (Primera Fase de la Experimentación)
- J48Consolidated con valores por defecto, muestras balanceadas y coverage = 99 % (Primera Fase de la Experimentación)

- Bagging: J48 y 100 muestras Bootstrap - (Primera Fase de la Experimentación)
- PCTBagging con valores por defecto, muestras balanceadas y coverage = 99 %
  - Porcentaje de Consolidación: 0, 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100
- PCTBagging con 100 muestras Bootstrap
- PCTRSM con valores por defecto, muestras balanceadas y coverage = 99 %, con un porcentaje de atributos del **50%**
- PCTRSM con valores por defecto, muestras balanceadas y coverage = 99 %, con un porcentaje de atributos del **75%**
- PCTRSM con  $N\_S = 100$  al modo RSM, sin remuestreo, con un porcentaje de atributos del **50%**
- PCTRSM con  $N\_S = 100$  al modo RSM, sin remuestreo, con un porcentaje de atributos del **75%**
- PCTRF con valores por defecto, muestras balanceadas y coverage = 99 %, con un porcentaje de atributos del **50%**
- PCTRF con valores por defecto, muestras balanceadas y coverage = 99 %, con un porcentaje de atributos del **75%**
- PCTRF con  $N\_S = 100$  al modo RSM, sin remuestreo, con un porcentaje de atributos del **50%**
- PCTRF con  $N\_S = 100$  al modo RSM, sin remuestreo, con un porcentaje de atributos del **75%**

Todos los algoritmos PCTEnsembles han sido evaluados con los siguientes porcentajes de consolidación: 0, 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100

Los criterios de bondad utilizados para este experimento no solo se basan en la capacidad de clasificación, sino que también se han añadido nuevos puntos de vista: capacidad explicativa y coste computacional.

- **Capacidad de clasificación:**
  - Tasa de acierto/Accuracy (Acc)
  - Balanced Accuracy
  - Kappa
  - Average area under ROC (AUC)
- **Capacidad explicativa** (En el caso de los PCTEnsembles, solo se mira la parte consolidada del árbol):
  - Número de nodos del árbol (TreeSize)
  - Número de nodos hoja del árbol (NumLeaves)
- **Coste computacional:**
  - Tiempo de construcción del clasificador (UserCPU Time Training)

## Capacidad de Clasificación

### Accuracy

En la primera gráfica, como se puede observar en la figura 80, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones con 100 muestras sin remuestreo son las que mejores resultados proporcionan con una notable diferencia. También hay una ligera mejora con el 50% de los atributos.

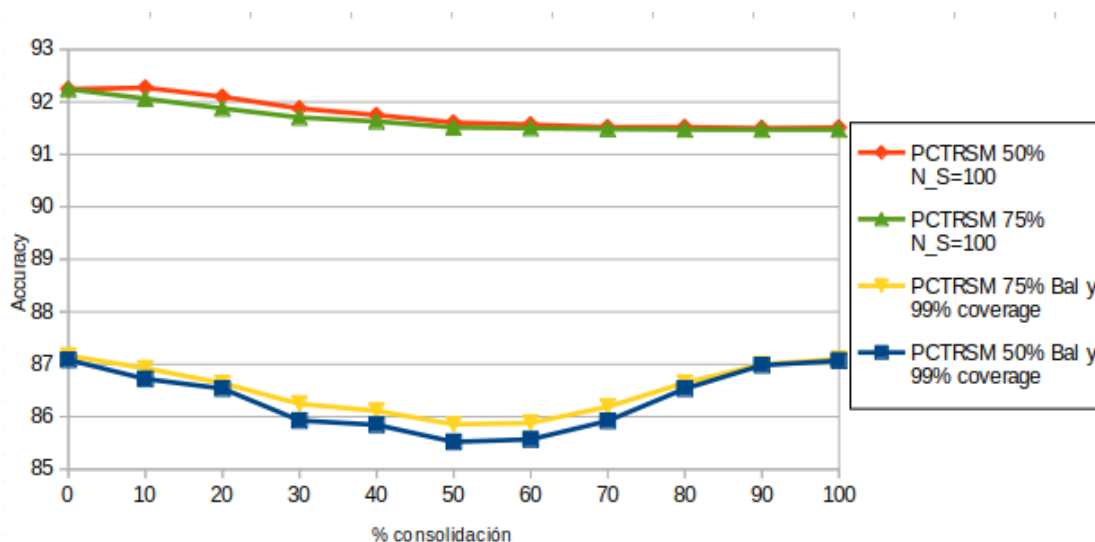


Figura 80: Gráfica Accuracy PCTRSM - Segunda Fase



Como se puede ver en la segunda gráfica, en la figura 81, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. Las versiones de 100 muestras sin remuestreo, tanto para PCTRSM como para PCTRF, son las que mejores resultados dan frente a las que usan muestras balanceadas. En los dos PCTEnsembles también resulta ligeramente mejor utilizar el 50% de atributos en vez de el 75%.

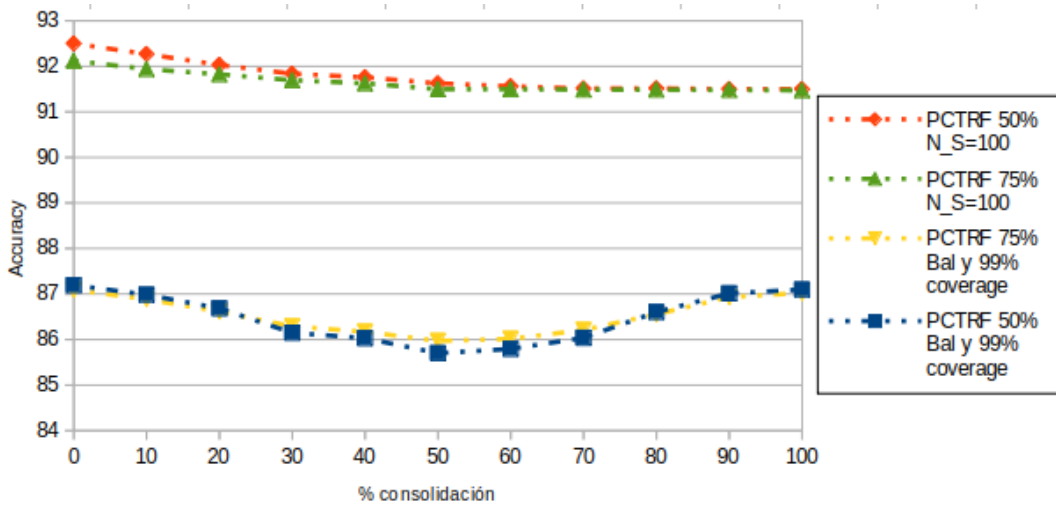


Figura 81: Gráfica Accuracy PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 82, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Se utiliza EL PCTBagging con 100 muestras Bootstrap debido a ser la versión con mejores resultados.

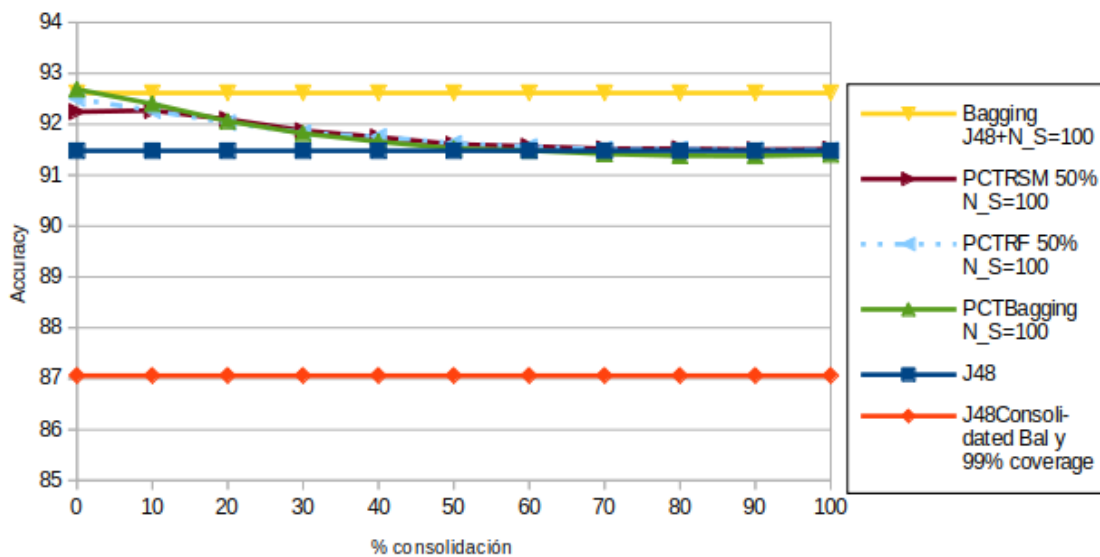


Figura 82: Gráfica Accuracy General - Segunda Fase

En la figura 83 se muestran los valores de los resultados para el Accuracy para verlos de una manera más clara:

	% Consolidación	0	10	20	30	40	50	60	70	80	90	100
Bagging J48+N	S=100	92,6157	92,6157	92,6157	92,6157	92,6157	92,6157	92,6157	92,6157	92,6157	92,6157	92,6157
PCTRSM 50%	N_S=100	92,2448	92,2718	92,0982	91,8749	91,7486	91,607	91,5683	91,5194	91,5221	91,5012	91,5125
PCTRF 50%	N_S=100	92,4997	92,2682	92,0297	91,8359	91,7576	91,6255	91,5623	91,5101	91,5147	91,4989	91,4971
PCTBagging	N_S=100	92,6858	92,4009	92,0618	91,8195	91,6588	91,524	91,4854	91,4184	91,3824	91,3788	91,4051
J48		91,4791	91,4791	91,4791	91,4791	91,4791	91,4791	91,4791	91,4791	91,4791	91,4791	91,4791
J48Consolidated	Bal y 99% coverage	87,0624	87,0624	87,0624	87,0624	87,0624	87,0624	87,0624	87,0624	87,0624	87,0624	87,0624

Figura 83: Tabla de valores del Accuracy

Para el Accuracy, el J48Consolidated obtiene los peores resultados. El PCTRSM, PCTRF y PCTBagging con valores muy similares, consiguen superar al J48 cuando el porcentaje de consolidación es inferior al 50%, pero se igualan a medida que aumenta. El Bagging obtiene los mejores resultados de todos los algoritmos sin duda, a pesar de que con un 0% de consolidación, los PCTEnsembles consiguen acercarse.

## Balanced Accuracy

En la primera gráfica, como se puede observar en la figura 84, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones con muestras balanceadas y un coverage del 99% obtienen los mejores resultados sin ninguna duda. También hay una leve mejoría con el 75 % de los atributos.

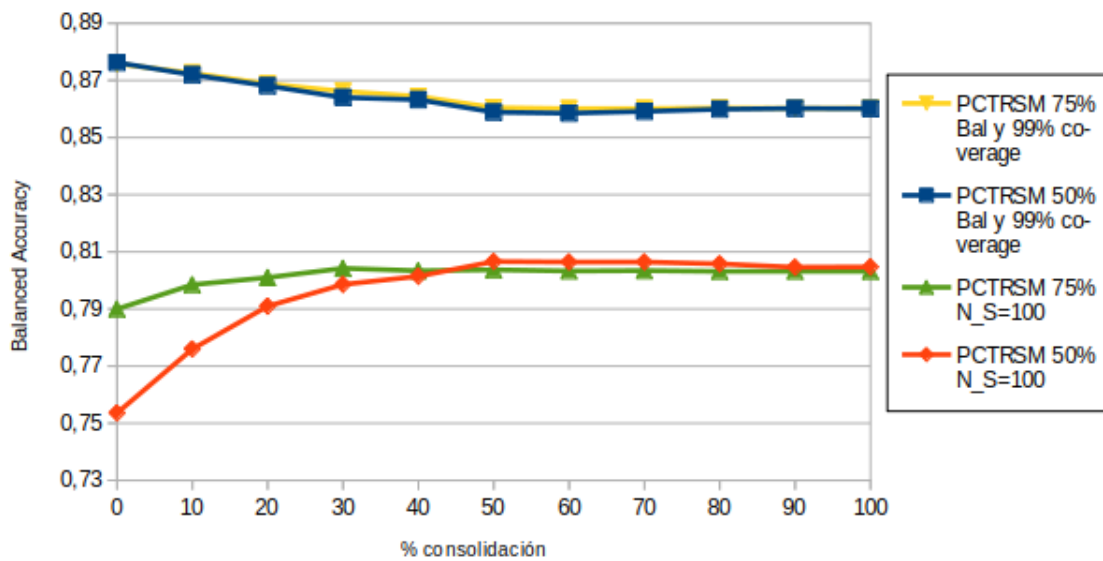


Figura 84: Gráfica Balanced Accuracy PCTRSM - Segunda Fase

Como se puede ver en la segunda gráfica, en la figura 85, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. Las versiones de muestras balanceadas y 99% de coverage, tanto para PCTRSM y PCTRF, son las que mejores resultados dan frente a las muestras sin remuestreo a diferencia del Accuracy. Para el PCTRF resulta ligeramente mejor el 50% de los atributos.

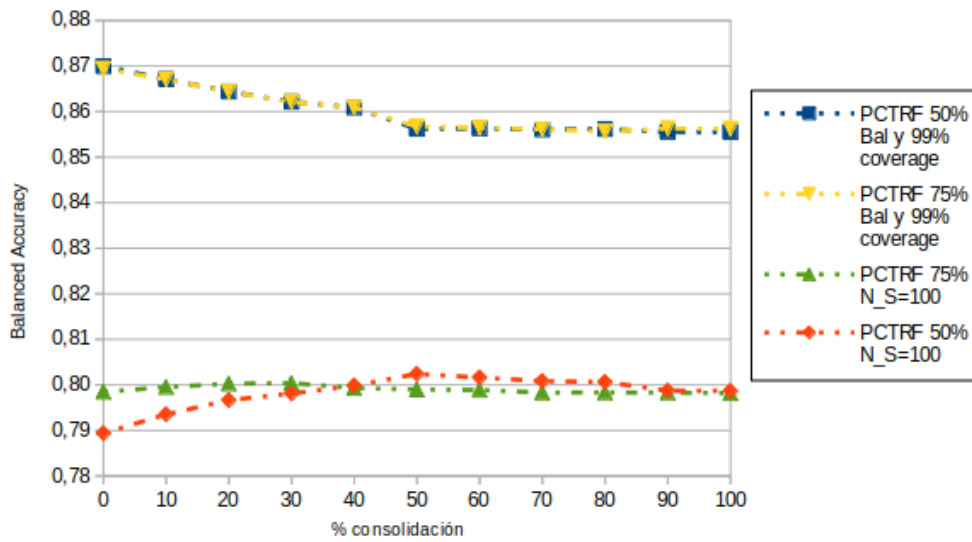


Figura 85: Gráfica Balanced Accuracy PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 86, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Se utiliza en PCTBagging con muestras balanceadas y un coverage del 99% debido a ser la versión con mejores resultados.

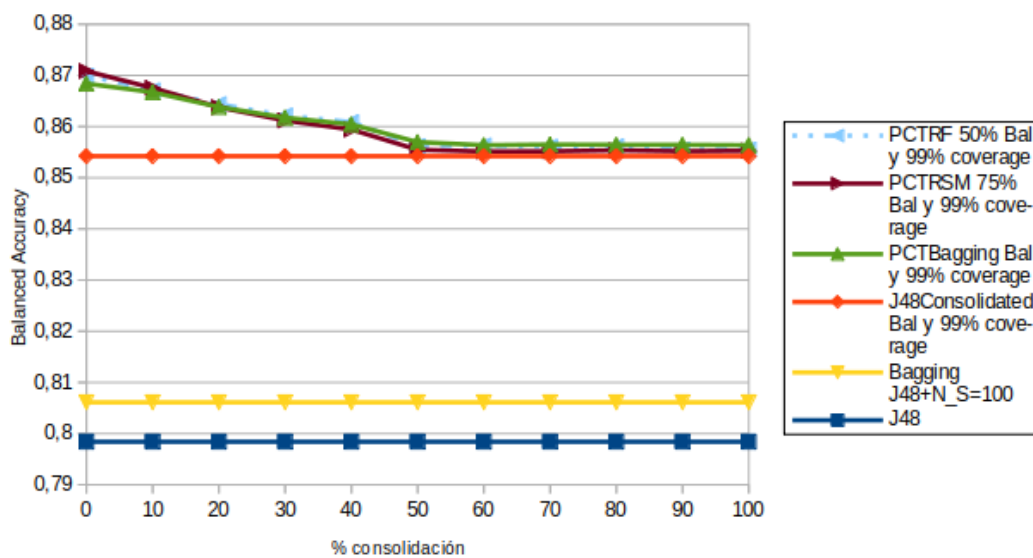


Figura 86: Gráfica Balanced Accuracy General - Segunda Fase

En la figura 87 se muestran los valores de los resultados para el Balanced Accuracy para verlos de una manera más clara:

	% Consolidación	0	10	20	30	40	50	60	70	80	90	100
PCTRF 50% Bal y 99% coverage		0,8700	0,8671	0,8644	0,8622	0,8609	0,8562	0,8563	0,8561	0,8562	0,8555	0,8555
PCTRSM 75% Bal y 99% coverage		0,8708	0,8675	0,8638	0,8612	0,8594	0,8555	0,8551	0,8552	0,8554	0,8552	0,8553
PCTBagging Bal y 99% coverage		0,8684	0,8667	0,8638	0,8617	0,8604	0,8570	0,8564	0,8565	0,8564	0,8564	0,8564
J48Consolidated Bal y 99% coverage		0,8542	0,8542	0,8542	0,8542	0,8542	0,8542	0,8542	0,8542	0,8542	0,8542	0,8542
Bagging J48+N_S=100		0,8061	0,8061	0,8061	0,8061	0,8061	0,8061	0,8061	0,8061	0,8061	0,8061	0,8061
J48		0,7984	0,7984	0,7984	0,7984	0,7984	0,7984	0,7984	0,7984	0,7984	0,7984	0,7984

Figura 87: Tabla de valores del Balanced Accuracy

Para el Balanced Accuracy, el J48, y Bagging se colocan con diferencia en la parte inferior de la gráfica. En esta medida los PCTEnsembles obtienen los mejores resultados, siendo el PCTRF el mejor de ellos por una mínima diferencia, superando al PCTRSM en prácticamente todos los porcentajes y al PCTBagging en los porcentajes inferiores. A medida que el porcentaje de consolidación aumenta los tres PCTEnsembles se igualan con el J48Consolidated, que cuando usan un 50% de consolidación o más, consigue unos resultados muy similares a estos.

## AUC

En la primera gráfica, como se puede observar en la figura 88, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones con muestras balanceadas y un coverage del 99% obtienen los mejores resultados de una manera clara. El 75% de los atributos también proporciona unos resultados levemente superiores.

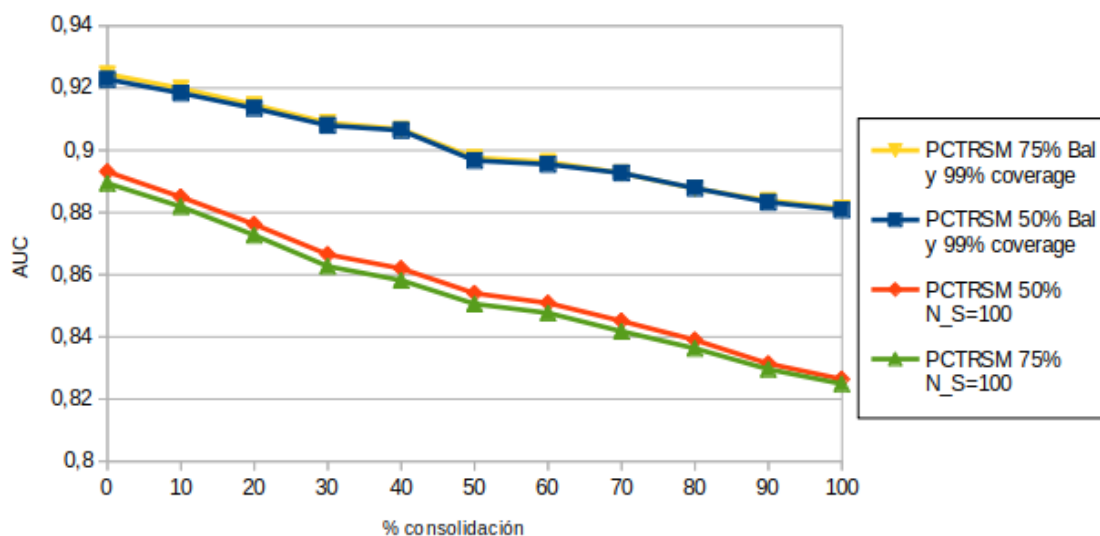


Figura 88: Gráfica AUC PCTRSM - Segunda Fase

Como se puede ver en la segunda gráfica, en la figura 89, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. Las versiones de muestras balanceadas y 99% de coverage, tanto para PCTRSM y PCTRF, son las que mejores resultados dan frente a las muestras sin remuestreo, como ocurre para el Balanced Accuracy. Para el PCTRF resulta ligeramente mejor el 50% de los atributos..

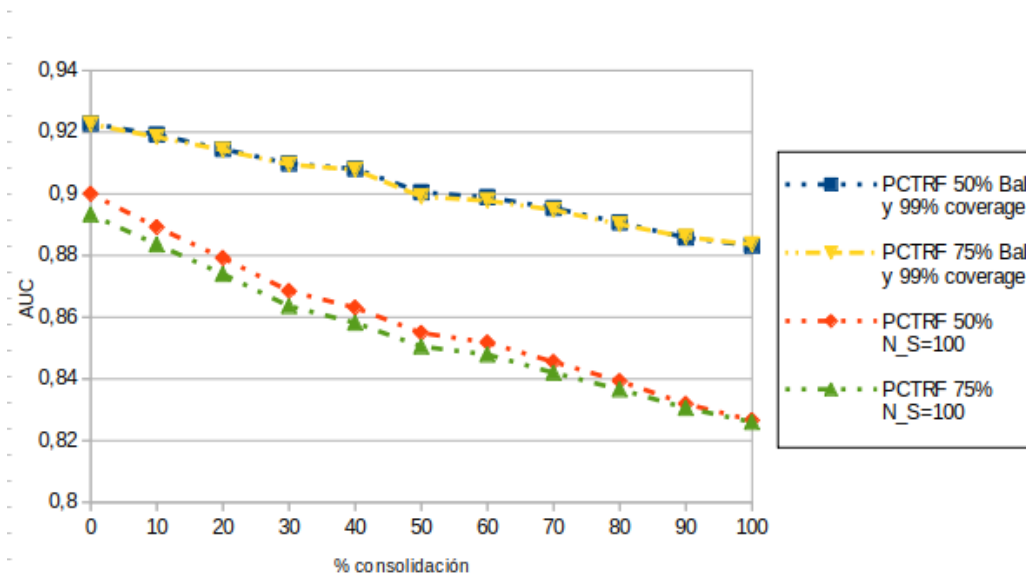


Figura 89: Gráfica AUC PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 90, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Se utiliza en PCTBagging con 100 muestras Bootstrap debido a ser la versión con mejores resultados.

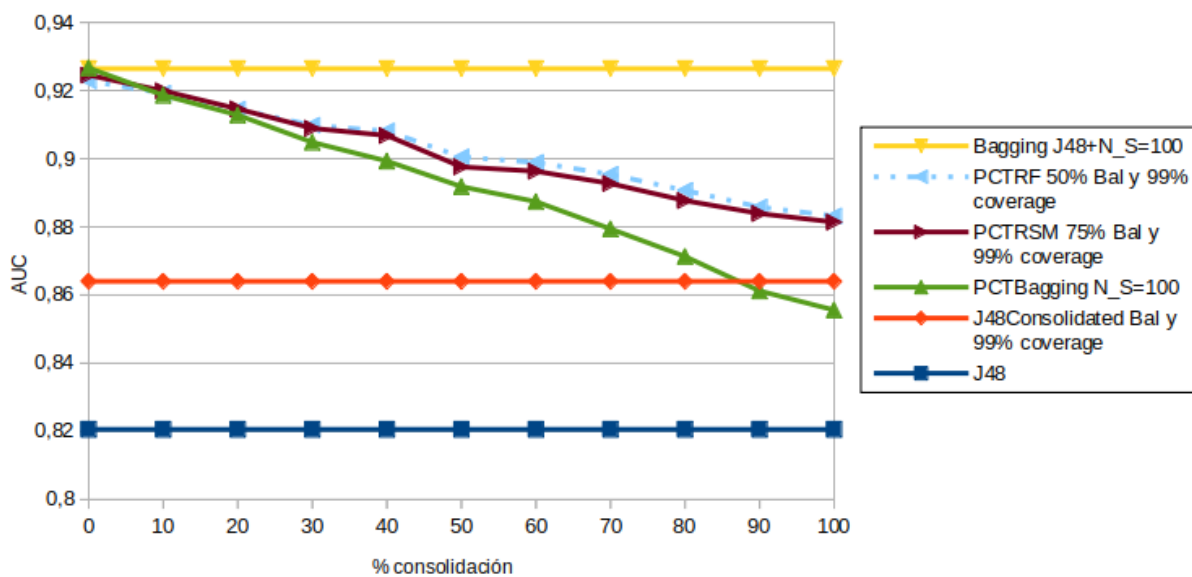


Figura 90: Gráfica AUC General - Segunda Fase

Para el AUC, el J48 obtiene los peores resultados de todos los algoritmos seguido del J48Consolidated. El PCTRF y el PCTRSM consiguen superar al PCTBagging en esta medida con valores superiores al 30% de consolidación, que aunque con los porcentajes de consolidación menores consiga igualarlos, a medida que va aumentando se va alejando. El Bagging se coloca en la parte superior siendo igualado solamente con porcentajes de consolidación del 0%.

## Kappa

En la primera gráfica, como se puede observar en la figura 91, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones con 100 muestras sin remuestreo son las que mejores resultados proporcionan con una notable diferencia. También hay una ligera mejora con el 50% de los atributos.

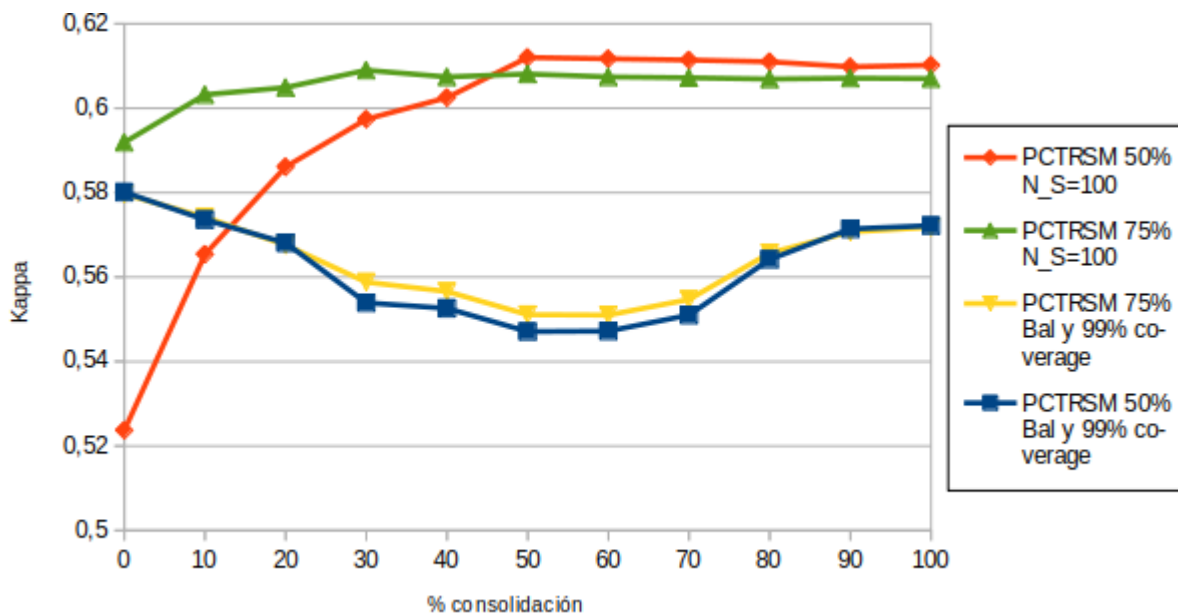


Figura 91: Gráfica Kappa PCTRSM - Segunda Fase

Como se puede ver en la segunda gráfica, en la figura 92, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. Las versiones de 100 muestras sin remuestreo, tanto para PCTRSM como para PCTRF, son las que mejores resultados dan frente a las que usan muestras balanceadas. En ambos PCTEnsembles también resulta ligeramente mejor utilizar el 50% de atributos en vez de el 75%.

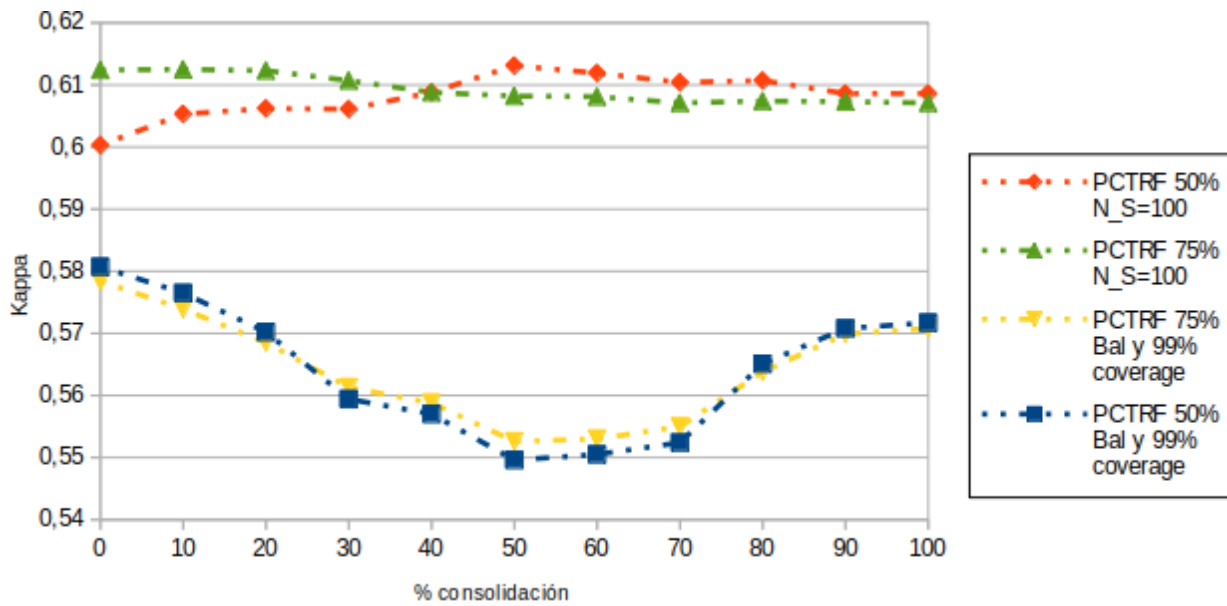


Figura 92: Gráfica Kappa PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 93, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Se utiliza en PCTBagging con 100 muestras Bootstrap debido a ser la versión con mejores resultados.

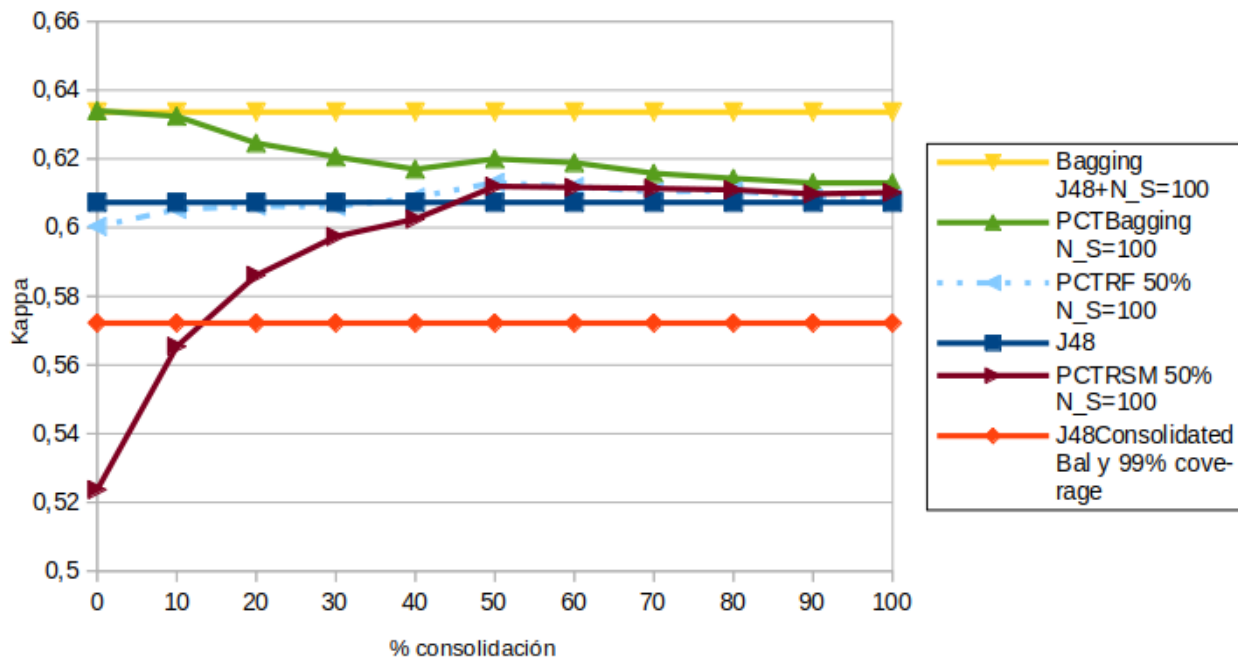


Figura 93: Gráfica Kappa General - Segunda Fase

Para Kappa, Bagging obtiene los mejores resultados de todos los algoritmos seguido del PCTBagging. El PCTRF se mantiene muy similar al J48 en todos los porcentajes de consolidación, a diferencia del PCTRSM que mejora a medida que el porcentaje aumenta, igualándose a partir del 50% de consolidación. El J48Consolidated, a pesar de mejorar al PCTRSM con un 0% y 10% de consolidación, se sitúa en la parte inferior con los peores resultados.

## Capacidad Explicativa

En estas medidas, los algoritmos que requieran de un menor número de nodos serán los mejores, ya que el objetivo es aportar una explicación a la clasificación de la manera más sencilla y fácil de comprender para el usuario.

## Número de nodos

En la primera gráfica, como se puede observar en la figura 94, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones con muestras balanceadas y un coverage del 99% son las que mejores resultados proporcionan con una clara diferencia.

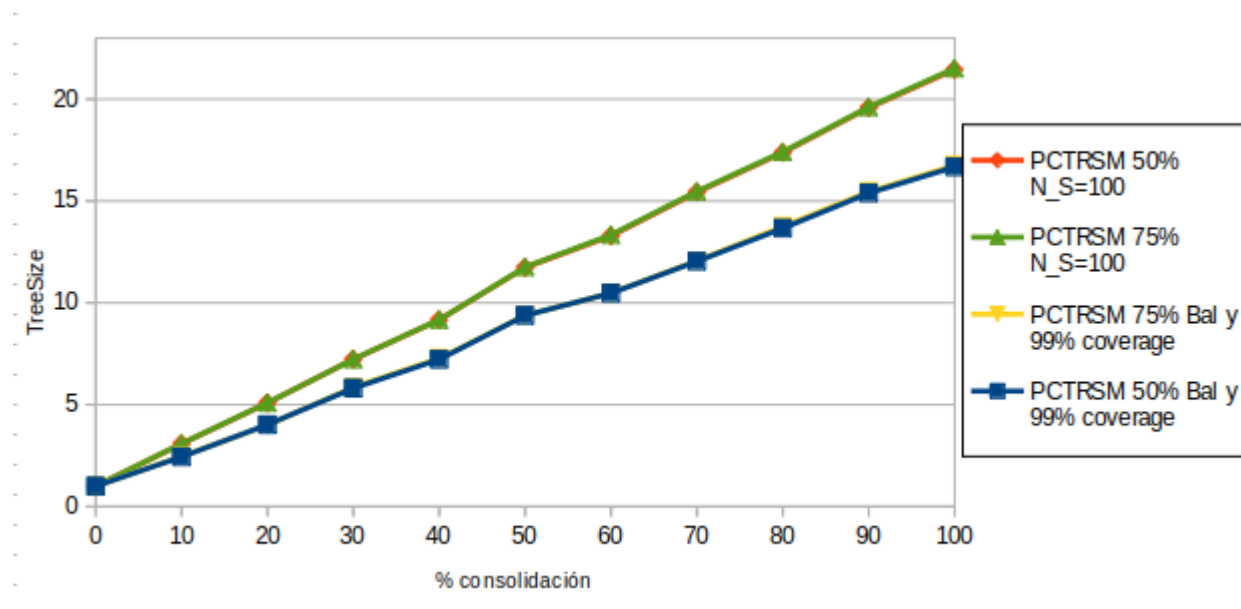


Figura 94: Gráfica Número de nodos PCTRSM - Segunda Fase



Como se puede ver en la segunda gráfica, en la figura 95, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. Las versiones de muestras balanceadas con un coverage del 99%, tanto para PCTRSM como para PCTRF, son las que mejores resultados dan frente a las que usan muestras sin remuestreo. Aunque en las gráficas no se pueda apreciar correctamente debido a los valores prácticamente iguales entre versiones con el mismo tipo de muestras, para ambos PCTEnsembles el 50% de los atributos proporciona una mínima mejoría.

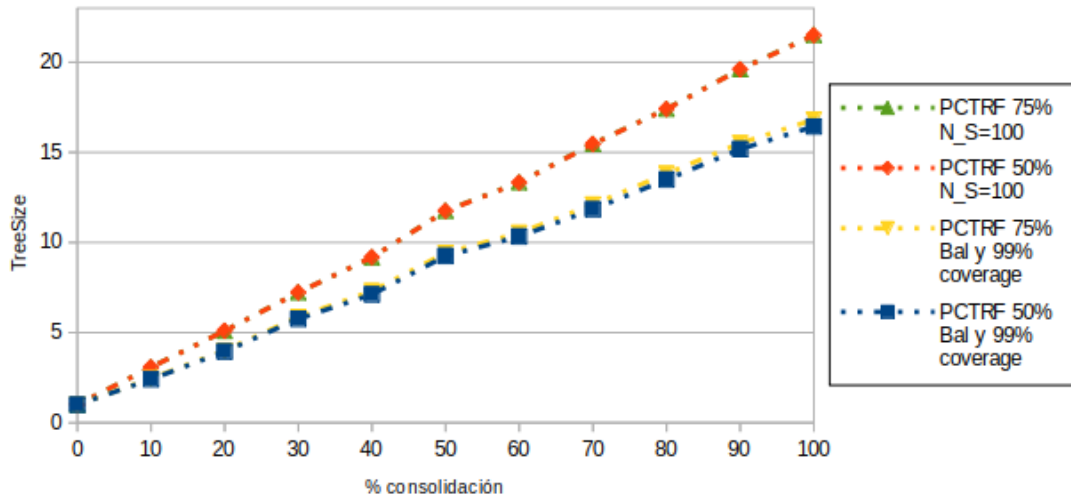


Figura 95: Gráfica Número de nodos PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 96, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Se utiliza en PCTBagging muestras balanceadas y un coverage del 99% debido a ser la versión con mejores resultados.

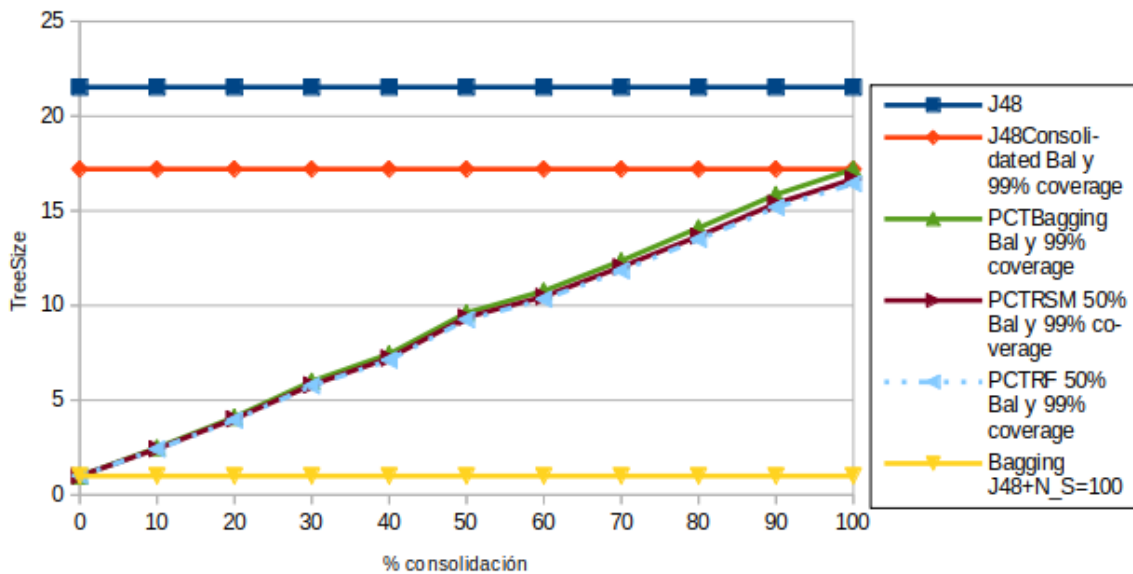


Figura 96: Gráfica Número de nodos General - Segunda Fase

En la figura 97 se muestran los valores de los resultados para el número de nodos del árbol para verlos de una manera más clara:

		% Consolidación	0	10	20	30	40	50	60	70	80	90	100
J48			21,5309	21,5309	21,5309	21,5309	21,5309	21,5309	21,5309	21,5309	21,5309	21,5309	21,5309
J48Consolidated	Bal y 99% coverage		17,2006	17,2006	17,2006	17,2006	17,2006	17,2006	17,2006	17,2006	17,2006	17,2006	17,2006
PCTBagging	Bal y 99% coverage		1	2,4897	4,103	5,9988	7,443	9,6145	10,7691	12,3606	14,1012	15,8588	17,2006
PCTRSM 50%	Bal y 99% coverage		1	2,4376	4,0182	5,8061	7,2248	9,38	10,4721	12,0406	13,6648	15,3891	16,6752
PCTRF 50%	Bal y 99% coverage		1	2,4109	3,9491	5,7582	7,1158	9,2509	10,3297	11,8388	13,4885	15,1745	16,4327
Bagging J48+N_S=100			1	1	1	1	1	1	1	1	1	1	1

Figura 97: Tabla de valores del número de nodos del árbol

En cuanto al número de nodos, Bagging obtendría el peor de los resultados, ya que al generar un árbol distinto para cada muestra, no aporta ninguna explicación al resultado. Entre los algoritmos PCT, el PCTRF obtiene los mejores resultados con el menor número de nodos para los árboles, seguido del PCTRSM, con el PCTBagging como el último. Tanto el PCTRF y el PCTRSM, consiguen mejorar al J48Consolidated en un 100% de consolidación. En último lugar se coloca el J48, con el mayor número de nodos.

### Número de nodos hoja

En la primera gráfica, como se puede observar en la figura 98, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones con muestras balanceadas y un coverage del 99% son las que mejores resultados proporcionan con una diferencia notable.

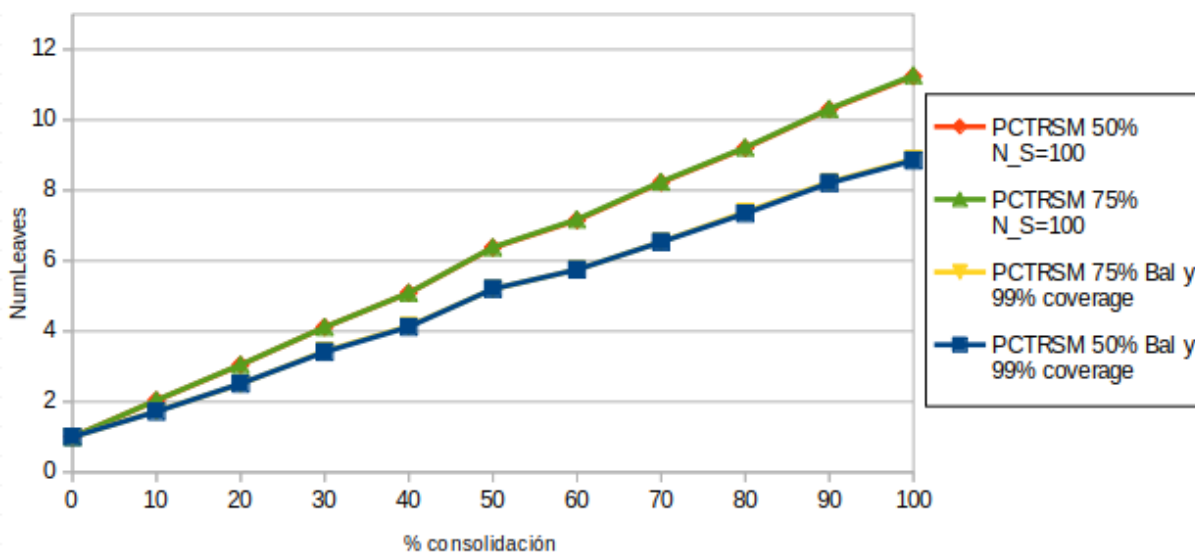


Figura 98: Gráfica Número de nodos hoja PCTRSM - Segunda Fase

Como se puede ver en la segunda gráfica, en la figura 99, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. La versión que mejores resultados proporciona para el número de nodos sería el PCTRF con muestras balanceadas y un coverage del 99%, tanto para PCTRSM como para PCTRF, son las que mejores resultados dan frente a las que usan muestras sin remuestreo. Aunque en las gráficas no se pueda apreciar correctamente debido a los valores prácticamente iguales entre versiones con el mismo tipo de muestras, para ambos PCTEnsembles el 50% de los atributos proporciona una mínima mejoría.

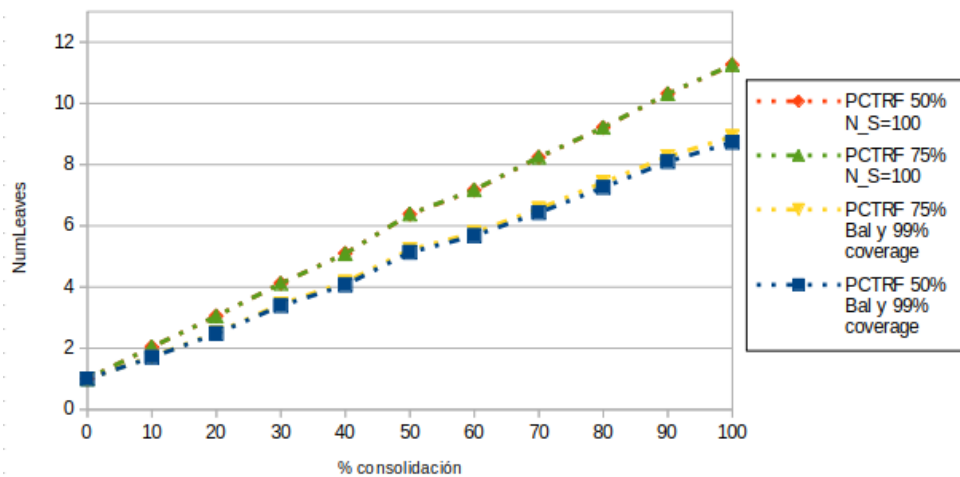


Figura 99: Gráfica Número de nodos hoja PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 100, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Se utiliza en PCTBagging muestras balanceadas y un coverage del 99% debido a ser la versión con mejores resultados.

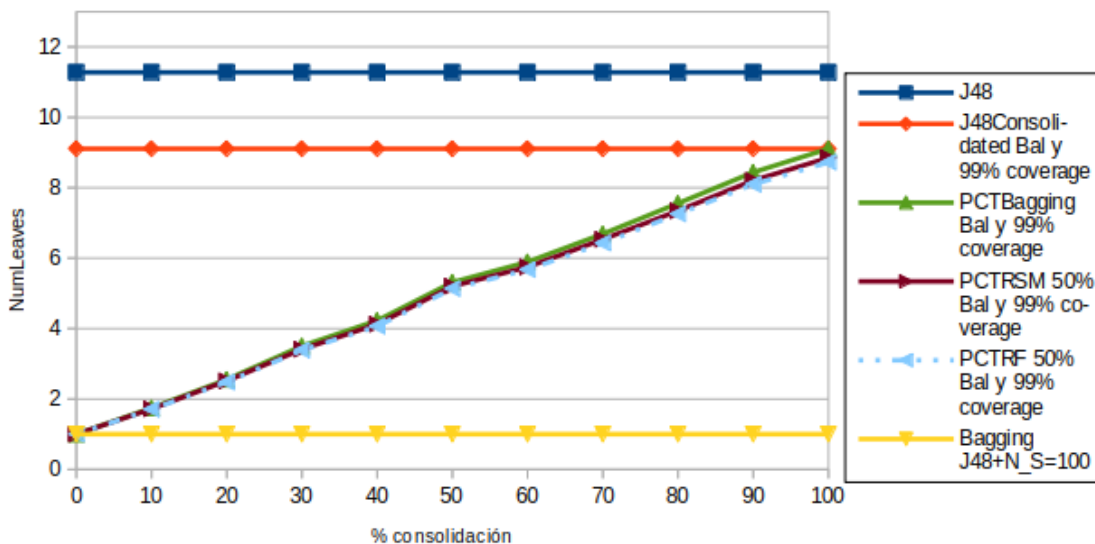


Figura 100: Gráfica Número de nodos hoja General - Segunda Fase

En la figura 101 se muestran los valores de los resultados para el número de nodos hoja del árbol para verlos de una manera más clara:

		% Consolidación	0	10	20	30	40	50	60	70	80	90	100
J48			11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752
J48Consolidated	Bal y 99% coverage		9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073
PCTBagging	Bal y 99% coverage		1	1,7448	2,5533	3,5048	4,2273	5,3139	5,8915	6,6873	7,5576	8,4364	9,1073
PCTRSM 50%	Bal y 99% coverage		1	1,7188	2,5133	3,4133	4,1236	5,2024	5,7485	6,5327	7,3448	8,2073	8,8503
PCTRF 50%	Bal y 99% coverage		1	1,7055	2,4776	3,3855	4,0648	5,1333	5,6727	6,4273	7,2521	8,0958	8,7248
Bagging J48+N	S=100		1	1	1	1	1	1	1	1	1	1	1

Figura 101: Tabla de valores del número de nodos hoja del árbol

En cuanto al número de nodos hoja, se repite la misma situación que para el número de nodos. PCTRF consigue mejorar al resto de los algoritmos con explicación, seguido del PCTRSM y PCTBagging. Bagging no nos aporta ninguna explicación y el J48 se coloca en último lugar.

## Coste Computacional

### Tiempo de construcción del clasificador

En la primera gráfica, como se puede observar en la figura 102, se comparan las 4 versiones posibles del PCTRSM en función del porcentaje de consolidación. Las versiones que mejores resultados proporcionan para el coste computacional son aquellas con muestras balanceadas y un coverage del 99%. También hay que resaltar, que como era de esperar, usar el 50% de atributos frente al 75% siempre reduce el tiempo de construcción del clasificador.

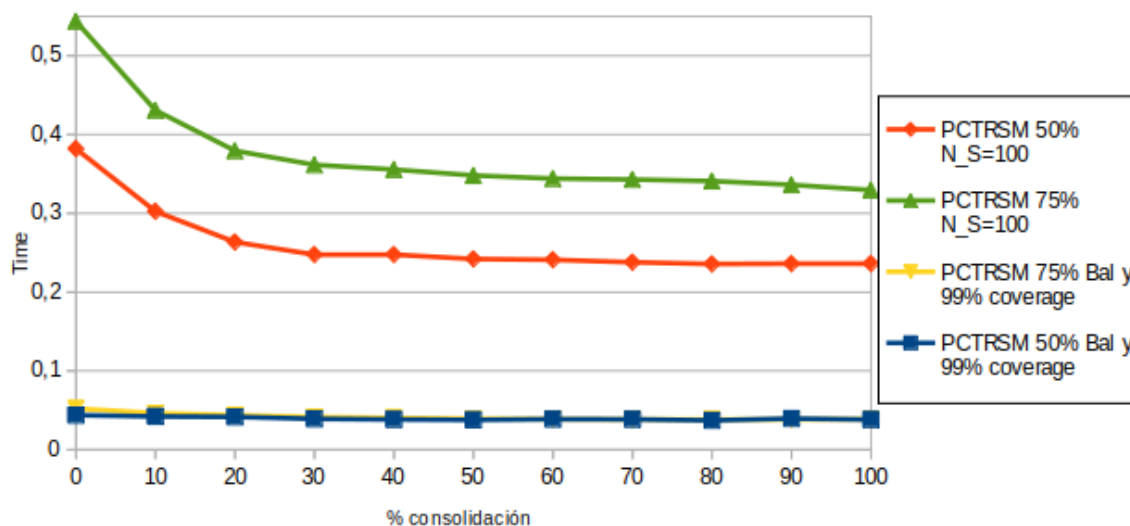


Figura 102: Gráfica Tiempo de construcción del clasificador PCTRSM - Segunda Fase

Como se puede ver en la segunda gráfica, en la figura 103, ahora se comparan las 4 versiones posibles del PCTRF con las mismas opciones que en el PCTRSM. De la misma manera que para PCTRSM, las versiones que mejores resultados proporcionan para el coste computacional son aquellas con muestras balanceadas y un coverage del 99%, al igual que la mejora que supone el 50% de los atributos ante el 75%.

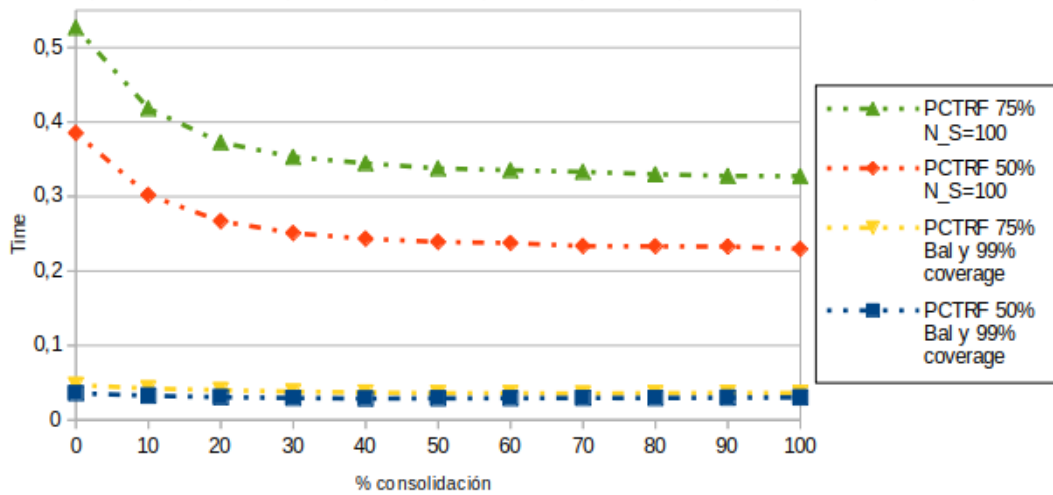


Figura 103: Gráfica Tiempo de construcción del clasificador PCTRF - Segunda Fase

Por último en la tercera gráfica, ver figura 104, se toman los resultados obtenidos de la mejor versión del PCTRSM y PCTRF para compararlos con el resto de los competidores elegidos. Para el coste computacional, el PCTBagging con muestras balanceadas es la versión con los mejores resultados por lo que será la elegida para la comparación.

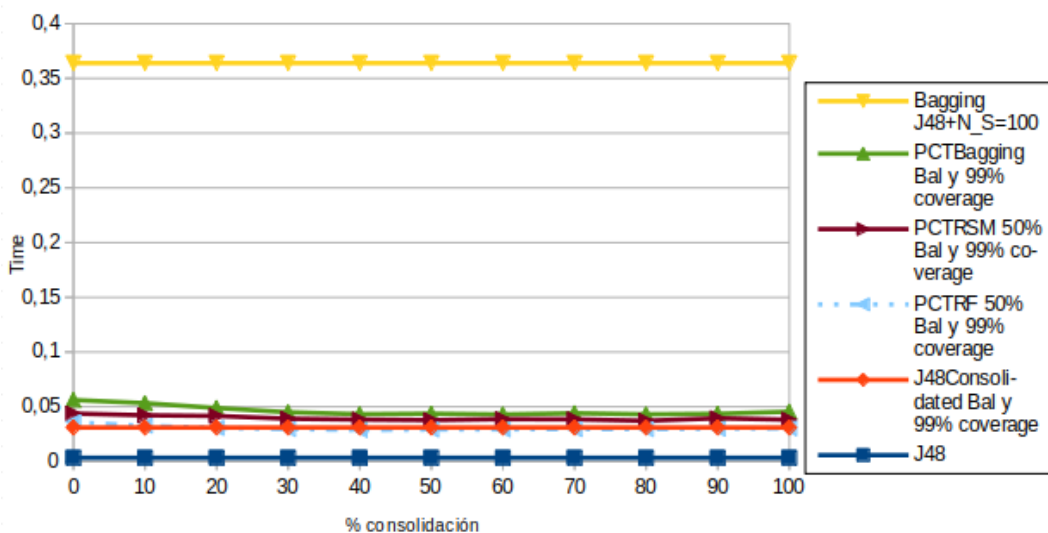


Figura 104: Gráfica Tiempo de construcción del clasificador General - Segunda Fase

Tal como se ve en la figura 105, se muestran los valores de los resultados para tiempo de construcción del clasificador para verlos de una manera más clara:

			% Consolidación	0	10	20	30	40	50	60	70	80	90	100
J48				11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752	11,2752
J48Consolidated	Bal y 99% coverage			9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073	9,1073
PCTBagging	Bal y 99% coverage			1	1,7448	2,5533	3,5048	4,2273	5,3139	5,8915	6,6873	7,5576	8,4364	9,1073
PCTRSM 50%	Bal y 99% coverage			1	1,7188	2,5133	3,4133	4,1236	5,2024	5,7485	6,5327	7,3448	8,2073	8,8503
PCTRF 50%	Bal y 99% coverage			1	1,7055	2,4776	3,3855	4,0648	5,1333	5,6727	6,4273	7,2521	8,0958	8,7248
Bagging J48+N_S=100				1	1	1	1	1	1	1	1	1	1	1

**Figura 105: Tabla de valores del tiempo de construcción del clasificador**

En cuanto al tiempo de construcción del clasificador, el J48 se sitúa en la mejor posición, debido a su sencillez en comparación al resto de algoritmos, con el menor tiempo, seguido del J48Consolidated. El PCTRF consigue los mejores resultados de los PCTEnsembles, seguido del PCTRSM y PCTBagging. Por último el Bagging, ya que no utiliza muestras balanceadas y coverage, si no 100 muestras bootstrap, en este caso.

### 9.2.1 CONCLUSIONES FASE 2

De nuevo, al igual que en la primera fase de la experimentación, los resultados varían en función al clasificador y los parámetros escogidos. En las medidas clasificatorias Accuracy y Kappa los algoritmos PCTRSM y PCTRF con muestras sin remuestreo consiguen los mejores valores, pero para el Balanced Accuracy y AUC las muestras balanceadas y coverage del 99% las superan. También se puede observar que entre un 50% y un 75% de los atributos parece no haber gran diferencia, aunque el 50% mejora levemente para la mayoría de las medidas, tanto para muestras balanceadas con coverage como muestras sin remuestreo.

Teniendo en cuenta que en esta experimentación se está trabajando con bases de datos desbalanceadas, se puede afirmar que el Balanced Accuracy y el AUC son las dos medidas clasificatorias más adecuadas, por lo que tanto PCTRSM como PCTRF con muestras balanceadas con un coverage del 99% serían las versiones con mejores resultados o con los resultados más relevantes. Esto se repite de igual manera para el PCTBagging.

Para el Balanced Accuracy, el PCTRF seguido del PCTRSM obtienen los mejores resultados, mejorando cualquiera de los otros algoritmos. En el resto de las medidas clasificatorias, aunque no se colocan en primer lugar, mejoran ligeramente al PCTBagging y consiguen el segundo y tercer lugar en todas ellas, exceptuando Kappa. Incluso para el AUC obtienen unos resultados notablemente mejores al PCTBagging a partir del 30% de consolidación, siendo este uno de los criterios más adecuados.

En cuanto a las medidas explicativas, tanto para el número de nodos como para el número de nodos hoja se repiten los resultados. Bagging, al no realizar ninguna consolidación de los árboles no aporta ninguna explicación de sus clasificaciones y no se toma en cuenta. A medida que el porcentaje de consolidación aumenta, el número de nodos también lo hace gradualmente. A mayor consolidación, mayor explicación, pero peores son sus capacidades clasificatorias para todas las medidas menos Kappa. Dependiendo del problema propuesto, se necesitará priorizar si es preferible la explicación o la mejor clasificación de los casos. Con un 0% de consolidación no se obtendrá ninguna explicación y con el 100% se podrá visualizar el árbol completo. En cualquier porcentaje de consolidación del árbol, el PCTRF depende de un ligero número menor de nodos, seguido del PCTRSM y PCTBagging, por lo que sería el mejor algoritmo en este aspecto.

Para el coste computacional, como era de esperar el J48 es el algoritmo que menos tiempo necesita, seguido por el J48Consolidated. Entre los PCTEnsembles, el PCTRF obtiene de nuevo los mejores resultados. El Bagging por su parte, debido a las muestra bootstrap frente a las balanceadas, requiere del mayor tiempo de entre todos ellos como se podía imaginar. También hay que resaltar que las muestras balanceadas mejoran este coste frente las muestras bootstrap e incluso las muestras sin remuestreo.

Como conclusión final, el PCTRF y el PCTRSM parecen obtener mejores resultados que el PCTBagging en prácticamente todas las medidas con una leve diferencia, siendo el PCTRF el mejor de ellos, debido a la aleatoriedad que aportan los subespacios. El PCTRF, que utiliza esta aleatoriedad en mayor medida que el PCTRSM, es dentro de estos algoritmos el que una mejor proporción de capacidad clasificatoria, capacidad explicativa y coste computacional ofrece.





## 10. CONCLUSIONES Y LÍNEAS ABIERTAS

En este proyecto se ha logrado diseñar e integrar los algoritmos PCTRSM y PCTRF diseñados por el grupo ALDAPA en la plataforma libre WEKA. El PCTBagging se diseñó con el objetivo de aumentar la explicatividad de las clasificaciones afectando lo menos posible a la capacidad clasificatoria de estas. El prefijo PCT viene de *Partially Consolidated Tree*, árboles parcialmente consolidados. El objetivo era crear un híbrido entre el CTC y el Bagging, dejando al usuario la capacidad de elegir la cantidad de explicación que se quiere obtener dependiendo de las circunstancias. Estos dos algoritmos, PCTRSM y PCTRF, nacen desde la misma perspectiva, basándose en las ideas del PCTBagging y añadiendo aleatoriedad al proceso, cada uno a su manera y con sus características, como el Random Subspace Method (RSM) y Random Forest (RF) respectivamente, comprobando así las diferencias que esto supone en los resultados.

Gracias a la experimentación que se ha llevado a cabo, se ha logrado asegurar la validez de la implementación y al mismo tiempo se han podido comparar los resultados con otros algoritmos considerados los competidores directos de estos, el PCTRSM y PCTRF.

Tras realizar una experimentación para obtener la mejor versión del RSM y el RF para posteriormente ser usados en el PCTRSM y el PCTRF, se han obtenido unos resultados muy interesantes en la segunda fase de esta experimentación. El uso de muestras balanceadas y un coverage del 99% ha resultado ser notablemente superior frente al uso de muestras sin remuestreo para las bases de datos desbalanceadas utilizadas. Estos dos algoritmos parecen conseguir una ligera mejora ante el PCTBagging en la mayoría de los criterios utilizados, ya sea para la capacidad clasificatoria, la capacidad explicativa o el coste computacional. La aleatoriedad que estos dos algoritmos aportan parece proporcionar una cierta mejora en las clasificaciones.

Por lo que a mí se refiere, este proyecto ha sido una magnífica experiencia. Me ha ayudado a conocer en primera persona lo que es enfrentarse a un proyecto de tales características. También ha sido una gran oportunidad para aprender el funcionamiento de Eclipse y WEKA a un nivel más interno, y ver cómo es el proceso de integrar un nuevo algoritmo en esta plataforma, así como también apreciar el trabajo que hicieron los desarrolladores de la universidad de Waikato al crear esta plataforma libre y gratuita que ayuda tanto a usuarios como a desarrolladores.

### 10.1 LÍNEAS ABIERTAS

Tras la realización del proyecto, quedan varias líneas abiertas:

- En las versiones de PCTRSM y PCTRF no se han usado muestras Bootstrap (como en el Bagging, PCTBagging y CTC), se podría ver qué resultados se obtienen con respecto a no usar remuestreo como en el RSM y RF. Para algunos criterios de bondad utilizados estos

tipos de muestras parecen dar buenos resultados.

- Implementar más PCTEnsembles. Cualquier ensemble que use árboles como clasificador base, podría ser usado. Un ejemplo sería hacer un Random Forest con el C4.5 Randomized. URL: <http://dx.doi.org/10.1023/A:1007607513941>. Hace lo mismo que Random Forest pero en lugar de elegir K atributos al azar en cada nodo y luego quedarse con el mejor de esos. Lo que hace es ordenar todos los atributos para dividir un nodo de mejor a peor y elegir uno al azar entre los K primeros.
- También se podría haber añadido a la experimentación otro PCTEnsemble como es el PCTBoosting, el TFG realizado por un compañero, Mikel González, aunque se ha que con la experimentación realizada era más que suficiente.
- En este sentido, también se podría ampliar la experimentación a otras bases de datos, por ejemplo los otros 2 conjuntos del repositorio KEEL, con bases de datos estándares (no desbalanceadas y no sólo biclásicas) y las bases de datos biclásicas usadas aquí pero procesadas con el SMOTE.
- Añadir las medidas explicativas a los ensembles (Bagging, RSM...), como la suma de esas medidas para cada árbol, aunque sean valores muy-muy altos y añadir en los PCTEnsembles una segunda versión de cada medida para medir, por ejemplo, el número de nodos hoja de todos los árboles (como si fuera el Bagging). La otra versión de la misma medida es la que tenemos ahora que sólo mide, por ejemplo, el número de nodos hoja de la parte consolidada del árbol. De esta manera podremos valorar mejor, cuantitativamente, cuál es la explicabilidad de un Ensemble y de un PCTEnsemble.

# 11. BIBLIOGRAFÍA

- [1] Igor Ibarguren, Jesús M. Pérez, Javier Muguerza, Olatz Arbelaitz and Ainhoa Yera. (2022); "PCTBagging: From Inner Ensembles to Ensembles. A trade-off between Discriminatin Capacity and Interpretability".
- [2] Jesús M. Pérez de la Fuente (2006); "Árboles Consolidados: Construcción de un árbol de clasificación basado en múltiples submuestras sin renunciar a la explicación".
- [3] Tin Kam Ho (1998); "The Random Subspace Method for Constructing Decision Forests"
- [4] Leo Breiman (1999); "Random Forests–Random Features"
- [5] Leo Breiman (1996); "Bagging Predictors"
- [6] Quinlan JR (1993) "C4.5: Programs for machine learning"
- [7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); "The WEKA Data Mining Software: An Update"



# 12. APÉNDICE: TABLAS DE DATOS DE LA EXPERIMENTACIÓN

## PRIMERA FASE

- Accuracy

Accuracy		J48		J48Consolidated				Bagging				
		defecto	defecto Balanced + Cov=99%	Bootstrap				defecto REP Tre + N_S=10	J48			
				N_S=10	N_S=30	N_S=50	N_S=100		N_S=10	N_S=30	N_S=50	N_S=100
Datasets		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
abalone19	(50)	99,2334	83,3543	83,92	82,7841	83,0765	82,7435	99,2334	99,2334	99,2334	99,2334	99,2334
abalone9-18	(50)	94,5694	77,8092	78,5345	78,714	77,6188	77,1809	95,117	95,4726	95,5683	95,7462	95,6364
ecoli-0_vs-1	(50)	98,5455	98,5455	98,5455	98,5455	98,5455	98,5455	98,7727	98,5	98,4091	98,4545	98,4545
ecoli1	(50)	89,7608	88,3889	87,7638	88,6585	89,1378	89,1352	90,4166	89,9996	90,327	90,3284	89,9706
ecoli2	(50)	93,7515	92,0843	91,4306	91,8464	91,935	92,2932	94,1383	94,3775	94,5562	94,7054	94,7643
ecoli3	(50)	92,4412	87,4965	87,0461	86,3916	87,1971	87,7371	92,7998	92,8907	93,2463	93,1576	93,2761
ecoli4	(50)	96,5487	91,8762	90,6308	91,7234	91,5751	91,7884	97,1431	97,4403	97,3222	97,2928	97,2634
glass-0-1-2-3_vs-4-5-6	(50)	92,7564	89,6334	89,68	89,5426	89,7265	89,3511	92,8505	92,887	93,1717	92,9369	93,0764
glass0	(50)	80,2713	79,5748	79,3433	79,577	79,0576	79,4784	83,9701	83,9291	84,7652	84,9014	85,4197
glass1	(50)	75,0354	72,2802	72,5559	72,9413	73,7353	73,6357	79,2392	78,3666	79,062	79,3477	80,6047
glass2	(50)	90,9812	73,8904	72,0919	72,938	74,0764	72,8859	92,0598	91,6899	91,4551	91,6877	91,8738
glass4	(50)	94,7265	89,4895	88,6434	88,6024	89,9535	89,7198	94,8195	95,4319	95,2913	95,1506	95,4795
glass5	(50)	98,7386	89,7475	88,5592	89,5947	90,0266	89,7475	96,3145	97,7132	97,9945	98,041	97,8549
glass6	(50)	94,948	93,2669	93,4064	93,1728	93,1262	93,1262	96,3488	95,2303	95,6999	95,6534	95,794
haberman1mb	(50)	70,982	66,3115	66,4109	65,5664	66,0502	66,2829	72,7118	71,412	71,8974	71,9942	71,8969
iris0	(50)	98,9333	98,9333	98,9333	98,9333	98,9333	98,9333	100	98,9333	98,9333	98,9333	98,9333
new-thyroid1	(50)	96,2791	96,6977	96,3721	96,9767	96,3721	96,4651	96,5581	97,3023	97,3953	97,3488	97,3023
new-thyroid2	(50)	96,4186	96,7442	95,907	96,7907	96,8837	96,8372	96,6047	96,7442	97,1628	97,1163	97,2093
page-blocks0	(50)	97,1107	95,2997	95,0804	95,2156	95,2632	95,2997	97,4598	97,4799	97,5658	97,5841	97,5786
pima1mb	(50)	73,2155	71,9014	71,6517	71,6134	72,1233	71,6543	76,0285	74,8019	75,7925	76,1702	76,3265
segment0	(50)	99,1939	98,7261	98,7391	98,7521	98,7347	98,7477	99,4193	99,3153	99,3023	99,3066	99,3196
vehicle0	(50)	94,1019	92,435	92,4123	92,5061	92,5541	92,5887	94,9179	95,0828	95,5204	95,8274	95,8391
vehicle0	(50)	76,1813	72,8715	73,2955	73,8279	74,0061	73,8758	77,5062	77,9896	78,6047	78,581	78,5936
vehicle0	(50)	95,8272	94,3835	94,3722	94,8336	95,0344	94,8213	96,891	97,5766	97,683	97,7182	97,7774
vehicle0	(50)	75,983	74,2084	74,2441	74,2224	74,0674	74,6581	78,5116	78,7127	78,8538	78,9009	78,9132
vowel0	(50)	98,8965	95,2423	95,09	95,0804	95,1612	94,9585	98,5124	98,7348	98,9573	98,9776	98,9877
wisconsin1mb	(50)	94,9056	94,9639	94,8462	94,8322	94,9198	94,6715	96,1199	96,1929	96,3979	96,4269	96,4562
yeast-2_vs-8	(50)	95,6632	87,7517	87,7992	87,1933	87,6314	87,1909	97,718	95,9749	95,9334	95,8918	95,8918
yeast1	(50)	75,0876	71,1454	71,1184	70,7814	70,8564	70,8831	76,772	76,7729	77,2243	77,5881	77,6557
yeast3	(50)	94,7033	93,895	93,922	93,8883	93,8614	93,8613	95,0268	94,9392	94,9122	94,8718	94,8381
yeast4	(50)	96,5228	79,7507	80,2623	79,6773	80,236	80,3564	96,8463	96,7387	96,8936	96,9543	97,0081
yeast5	(50)	98,5377	94,8586	94,3065	94,7371	94,6096	95,074	98,4635	98,5511	98,632	98,7331	98,7331
yeast6	(50)	97,9583	89,5015	89,8175	89,8037	89,6562	89,7976	98,2548	98,3693	98,3626	98,3626	98,3559
		91,4791	87,0624	86,8707	86,9777	87,1437	87,1008	92,3499	92,2663	92,4887	92,5432	92,6157
Average		91,4791	87,0624	86,8707	86,9777	87,1437	87,1008	92,3499	92,2663	92,4887	92,5432	92,6157



• **Balanced Accuracy**

Balanced Accuracy		J48		J48Consolidated				Bagging				
		defecto	defecto Balanced + Cov=99%	Bootstrap				defecto REPTree + N_S=10	J48			
				N_S=10	N_S=30	N_S=50	N_S=100		N_S=10	N_S=30	N_S=50	N_S=100
Datasets		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
abalone19	(50)	0,5000	0,6612	0,6319	0,6468	0,6515	0,6501	0,5000	0,5000	0,5000	0,5000	0,5000
abalone9-18	(50)	0,6310	0,7094	0,7046	0,7062	0,7053	0,7087	0,6026	0,6441	0,6468	0,6567	0,6482
ecoli-0_vs-1	(50)	0,9829	0,9829	0,9829	0,9829	0,9829	0,9829	0,9828	0,9826	0,9819	0,9822	0,9822
ecoli1	(50)	0,8437	0,8926	0,8864	0,8919	0,8962	0,8966	0,8589	0,8576	0,8629	0,8611	0,8516
ecoli2	(50)	0,8608	0,8762	0,8752	0,8742	0,8769	0,8768	0,8615	0,8639	0,8666	0,8730	0,8749
ecoli3	(50)	0,7508	0,8633	0,8684	0,8609	0,8705	0,8735	0,7402	0,7558	0,7654	0,7599	0,7656
ecoli4	(50)	0,8084	0,8842	0,8729	0,8764	0,8780	0,8838	0,8373	0,8576	0,8570	0,8568	0,8590
glass-0-1-2-3_vs-4-5-6	(50)	0,8941	0,8930	0,8948	0,8932	0,8918	0,8879	0,9059	0,8984	0,9020	0,8979	0,9008
glass0	(50)	0,7940	0,8001	0,7988	0,8002	0,7966	0,7968	0,8167	0,8157	0,8263	0,8288	0,8355
glass1	(50)	0,7089	0,7083	0,7064	0,7055	0,7175	0,7180	0,7499	0,7439	0,7492	0,7515	0,7644
glass2	(50)	0,5889	0,7337	0,7257	0,7134	0,7303	0,7272	0,5045	0,5223	0,5082	0,5185	0,5226
glass4	(50)	0,7604	0,8740	0,8756	0,8771	0,8796	0,8753	0,6282	0,7518	0,7292	0,7394	0,7507
glass5	(50)	0,9219	0,9464	0,9403	0,9456	0,9478	0,9464	0,6271	0,8687	0,8799	0,8706	0,8649
glass6	(50)	0,8918	0,9176	0,9184	0,9170	0,9168	0,9168	0,9205	0,9035	0,9170	0,9153	0,9175
haberman1mb	(50)	0,5719	0,6296	0,6294	0,6169	0,6166	0,6190	0,5800	0,5652	0,5650	0,5661	0,5690
iris0	(50)	0,9840	0,9840	0,9840	0,9840	0,9840	0,9840	1,0000	0,9840	0,9840	0,9840	0,9840
new-thyroid1	(50)	0,9260	0,9527	0,9507	0,9555	0,9496	0,9490	0,9219	0,9436	0,9430	0,9404	0,9368
new-thyroid2	(50)	0,9268	0,9541	0,9411	0,9544	0,9572	0,9581	0,9222	0,9345	0,9382	0,9368	0,9385
page-blocks0	(50)	0,9173	0,9497	0,9468	0,9507	0,9502	0,9505	0,9197	0,9251	0,9269	0,9267	0,9261
pima1mb	(50)	0,6992	0,7164	0,7108	0,7121	0,7125	0,7094	0,7218	0,7105	0,7236	0,7276	0,7279
segment0	(50)	0,9813	0,9840	0,9844	0,9841	0,9844	0,9842	0,9846	0,9832	0,9833	0,9831	0,9831
vehicle0	(50)	0,9225	0,9311	0,9322	0,9315	0,9319	0,9329	0,9359	0,9360	0,9427	0,9480	0,9479
vehicle0	(50)	0,6666	0,7377	0,7427	0,7435	0,7516	0,7463	0,6548	0,6793	0,6834	0,6821	0,6826
vehicle0	(50)	0,9451	0,9447	0,9436	0,9462	0,9478	0,9466	0,9593	0,9680	0,9684	0,9693	0,9701
vehicle0	(50)	0,6701	0,7551	0,7528	0,7459	0,7393	0,7482	0,6550	0,6775	0,6730	0,6699	0,6670
vowel0	(50)	0,9660	0,9608	0,9580	0,9600	0,9609	0,9603	0,9504	0,9576	0,9673	0,9659	0,9660
wisconsin1mb	(50)	0,9455	0,9509	0,9496	0,9488	0,9494	0,9472	0,9579	0,9599	0,9620	0,9626	0,9626
yeast-2_vs-8	(50)	0,5086	0,7328	0,7426	0,7371	0,7346	0,7299	0,7489	0,5294	0,5124	0,5074	0,5050
yeast1	(50)	0,6667	0,7026	0,7019	0,7030	0,7035	0,7066	0,6781	0,6840	0,6866	0,6891	0,6911
yeast3	(50)	0,8743	0,9276	0,9299	0,9278	0,9287	0,9279	0,8790	0,8684	0,8698	0,8701	0,8680
yeast4	(50)	0,6273	0,8193	0,8142	0,8166	0,8166	0,8196	0,5884	0,6252	0,6354	0,6357	0,6378
yeast5	(50)	0,8921	0,9581	0,9577	0,9543	0,9545	0,9537	0,8689	0,8704	0,8763	0,8870	0,8848
yeast6	(50)	0,7191	0,8542	0,8573	0,8586	0,8606	0,8558	0,6969	0,7338	0,7365	0,7393	0,7337
		<b>0,7984</b>	<b>0,8542</b>	<b>0,8519</b>	<b>0,8522</b>	<b>0,8538</b>	<b>0,8536</b>	<b>0,7927</b>	<b>0,8031</b>	<b>0,8051</b>	<b>0,8061</b>	<b>0,8066</b>
Average												





- TPR

TPR		J48		J48Consolidated				Bagging				
		defecto	defecto Balancec + Cov=99%	Bootstrap				defecto REPTree + N_S=10	J48			
				N_S=10	N_S=30	N_S=50	N_S=100		N_S=10	N_S=30	N_S=50	N_S=100
Datasets		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
abalone19	(50)	0	0,4862	0,4214	0,4629	0,4695	0,47	0	0	0	0	0
abalone9-18	(50)	0,2753	0,6317	0,6133	0,6147	0,625	0,6375	0,2086	0,2931	0,2978	0,3169	0,3
ecoli-0_vs-1	(50)	0,9916	0,9916	0,9916	0,9916	0,9916	0,9916	0,9993	0,9909	0,9895	0,9902	0,9902
ecoli1	(50)	0,744	0,9083	0,9021	0,9011	0,9047	0,9059	0,7752	0,7792	0,7883	0,7832	0,7627
ecoli2	(50)	0,7495	0,8116	0,8187	0,81	0,8155	0,81	0,7458	0,7482	0,7522	0,7658	0,7695
ecoli3	(50)	0,5314	0,8486	0,8657	0,8571	0,8686	0,8686	0,5029	0,5371	0,5543	0,5429	0,5543
ecoli4	(50)	0,63	0,845	0,835	0,83	0,835	0,845	0,685	0,725	0,725	0,725	0,73
glass-0-1-2-3_vs-4-5-6	(50)	0,8298	0,8864	0,8905	0,8887	0,8811	0,8769	0,8627	0,8398	0,8451	0,8375	0,8433
glass0	(50)	0,7686	0,8129	0,8143	0,8129	0,8143	0,8029	0,75	0,7471	0,7643	0,77	0,7814
glass1	(50)	0,5662	0,6593	0,6404	0,6233	0,6486	0,6549	0,6033	0,6064	0,6062	0,6064	0,6207
glass2	(50)	0,2067	0,7283	0,7317	0,695	0,7183	0,725	0,01	0,0533	0,025	0,045	0,0517
glass4	(50)	0,5467	0,85	0,8633	0,8667	0,8567	0,85	0,2633	0,52	0,4733	0,4967	0,5167
glass5	(50)	0,85	1	1	1	1	1	0,26	0,75	0,77	0,75	0,74
glass6	(50)	0,8127	0,8967	0,8967	0,8967	0,8967	0,8967	0,862	0,8367	0,862	0,8587	0,862
haberman1mb	(50)	0,2789	0,5588	0,5561	0,5346	0,5234	0,526	0,2675	0,2485	0,2379	0,2392	0,2503
iris0	(50)	0,968	0,968	0,968	0,968	0,968	0,968	1	0,968	0,968	0,968	0,968
new-thyroid1	(50)	0,8714	0,9314	0,9314	0,9343	0,9286	0,9257	0,8571	0,9	0,8971	0,8914	0,8829
new-thyroid2	(50)	0,8714	0,9343	0,9143	0,9343	0,94	0,9429	0,8571	0,8857	0,8886	0,8857	0,8886
page-blocks0	(50)	0,8496	0,9456	0,9417	0,9487	0,9472	0,9472	0,8507	0,8625	0,8655	0,8648	0,8637
pima1mb	(50)	0,5899	0,7077	0,6915	0,6988	0,6833	0,6855	0,5944	0,5862	0,61	0,6146	0,6105
segment0	(50)	0,9663	0,9793	0,9802	0,9793	0,9802	0,9796	0,9711	0,9693	0,9696	0,969	0,969
vehicle0	(50)	0,8874	0,9437	0,9473	0,9437	0,9437	0,9462	0,9106	0,9079	0,919	0,9286	0,9281
vehicle0	(50)	0,471	0,7562	0,7626	0,7542	0,7751	0,7617	0,408	0,4727	0,4727	0,469	0,4704
vehicle0	(50)	0,9178	0,9463	0,9431	0,9417	0,9426	0,9431	0,9395	0,9519	0,9509	0,9528	0,9541
vehicle0	(50)	0,4899	0,7812	0,7735	0,7535	0,7365	0,7513	0,3942	0,4576	0,4413	0,431	0,4221
vowel0	(50)	0,9378	0,9711	0,9667	0,9711	0,9722	0,9733	0,9078	0,9211	0,94	0,9367	0,9367
wisconsin1mb	(50)	0,9335	0,9548	0,9531	0,9502	0,9498	0,9485	0,9469	0,9531	0,9552	0,9569	0,9561
yeast-2_vs-8	(50)	0,02	0,575	0,595	0,59	0,58	0,575	0,5	0,06	0,025	0,015	0,01
yeast1	(50)	0,4671	0,6814	0,6797	0,6914	0,6914	0,7014	0,4657	0,4854	0,4835	0,4833	0,4884
yeast3	(50)	0,781	0,9129	0,9178	0,9135	0,916	0,9141	0,7876	0,7645	0,768	0,7693	0,7649
yeast4	(50)	0,2644	0,8427	0,8267	0,8378	0,832	0,8367	0,1802	0,2576	0,2771	0,2771	0,2809
yeast5	(50)	0,7928	0,9681	0,9731	0,9617	0,9633	0,9567	0,7458	0,7481	0,7594	0,7803	0,7758
yeast6	(50)	0,4457	0,8114	0,8143	0,8171	0,8229	0,8114	0,3971	0,4714	0,4771	0,4829	0,4714
----->		<b>0,6456</b>	<b>0,8341</b>	<b>0,8309</b>	<b>0,8295</b>	<b>0,8310</b>	<b>0,8312</b>	<b>0,6215</b>	<b>0,6454</b>	<b>0,6472</b>	<b>0,6486</b>	<b>0,6489</b>
Average		0,6456	0,8341	0,8309	0,8295	0,831	0,8312	0,6215	0,6454	0,6472	0,6486	0,6489
			(17/16/0)	(0/33/0)	(5/28/0)	(0/33/0)	(2/30/1)	(2/30/1)	(2/31/0)	(2/31/0)	(2/31/0)	(2/31/0)



• TNR

TNR		J48		J48Consolidated				Bagging				
		defecto	defecto	Bootstrap				defecto	J48			
			Balanced + Cov=99%	N_S=10	N_S=30	N_S=50	N_S=100	REPTree + N_S=10	N_S=10	N_S=30	N_S=50	N_S=100
Datasets	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	
abalone19	(50)	1	0,8362	0,8424	0,8306	0,8335	0,8302	1	1	1	1	1
abalone9-18	(50)	0,9866	0,7871	0,7958	0,7977	0,7855	0,7799	0,9965	0,9951	0,9958	0,9965	0,9964
ecoli-0_vs_1	(50)	0,9742	0,9742	0,9742	0,9742	0,9742	0,9742	0,9663	0,9742	0,9742	0,9742	0,9742
ecoli1	(50)	0,9433	0,8768	0,8707	0,8827	0,8877	0,8873	0,9425	0,9359	0,9374	0,939	0,9405
ecoli2	(50)	0,9721	0,9408	0,9317	0,9384	0,9383	0,9436	0,9771	0,9795	0,9809	0,9802	0,9802
ecoli3	(50)	0,9701	0,878	0,871	0,8647	0,8724	0,8784	0,9774	0,9745	0,9764	0,9768	0,9768
ecoli4	(50)	0,9867	0,9234	0,9108	0,9228	0,9209	0,9225	0,9895	0,9902	0,9889	0,9886	0,988
glass-0-1-2-3_vs_4-5-6	(50)	0,9583	0,8995	0,899	0,8977	0,9025	0,8988	0,9491	0,957	0,9589	0,9583	0,9583
glass0	(50)	0,8194	0,7873	0,7832	0,7874	0,7789	0,7907	0,8834	0,8842	0,8882	0,8875	0,8896
glass1	(50)	0,8515	0,7572	0,7723	0,7877	0,7863	0,781	0,8965	0,8814	0,8922	0,8965	0,9081
glass2	(50)	0,971	0,7391	0,7196	0,7318	0,7422	0,7293	0,999	0,9913	0,9914	0,9919	0,9934
glass4	(50)	0,9741	0,898	0,8879	0,8875	0,9025	0,9005	0,993	0,9836	0,9851	0,9821	0,9846
glass5	(50)	0,9937	0,8927	0,8805	0,8912	0,8956	0,8927	0,9941	0,9873	0,9898	0,9912	0,9898
glass6	(50)	0,9708	0,9384	0,94	0,9373	0,9368	0,9368	0,9789	0,9703	0,9719	0,9719	0,973
haberman1mb	(50)	0,8649	0,7004	0,7027	0,6991	0,7098	0,712	0,8924	0,8818	0,892	0,8929	0,8876
iris0	(50)	1	1	1	1	1	1	1	1	1	1	1
new-thyroid1	(50)	0,9806	0,9739	0,97	0,9767	0,9706	0,9722	0,9867	0,9872	0,9889	0,9894	0,9906
new-thyroid2	(50)	0,9822	0,9739	0,9678	0,9744	0,9744	0,9733	0,9872	0,9833	0,9878	0,9878	0,9883
page-blocks0	(50)	0,9849	0,9538	0,9518	0,9526	0,9532	0,9537	0,9887	0,9876	0,9882	0,9885	0,9885
pima1mb	(50)	0,8084	0,725	0,73	0,7254	0,7416	0,7332	0,8492	0,8348	0,8372	0,8406	0,8452
segment0	(50)	0,9962	0,9886	0,9886	0,9889	0,9885	0,9888	0,998	0,9971	0,9969	0,9971	0,9972
vehicle0	(50)	0,9575	0,9184	0,917	0,9193	0,92	0,9196	0,9611	0,964	0,9663	0,9674	0,9677
vehicle1	(50)	0,8621	0,7192	0,7228	0,7327	0,728	0,7308	0,9016	0,8858	0,8941	0,8951	0,8947
vehicle2	(50)	0,9723	0,943	0,944	0,9506	0,953	0,95	0,9791	0,9841	0,9858	0,9857	0,986
vehicle3	(50)	0,8502	0,729	0,732	0,7383	0,7421	0,745	0,9158	0,8973	0,9046	0,9087	0,9118
vowel0	(50)	0,9941	0,9505	0,9493	0,9488	0,9495	0,9472	0,9929	0,994	0,9945	0,9951	0,9952
wisconsin1mb	(50)	0,9574	0,9469	0,946	0,9473	0,9489	0,9458	0,9689	0,9667	0,9687	0,9682	0,9691
yeast-2_vs_8	(50)	0,9972	0,8906	0,8902	0,8841	0,8891	0,8848	0,9978	0,9987	0,9998	0,9998	1
yeast1	(50)	0,8663	0,7237	0,724	0,7145	0,7155	0,7118	0,8905	0,8826	0,8897	0,8949	0,8937
yeast3	(50)	0,9675	0,9422	0,9419	0,942	0,9414	0,9416	0,9703	0,9722	0,9715	0,9709	0,971
yeast4	(50)	0,9902	0,7959	0,8017	0,7953	0,8012	0,8024	0,9965	0,9927	0,9936	0,9942	0,9946
yeast5	(50)	0,9913	0,948	0,9422	0,9469	0,9456	0,9506	0,9919	0,9927	0,9932	0,9936	0,9937
yeast6	(50)	0,9925	0,897	0,9002	0,9	0,8983	0,9001	0,9967	0,9961	0,9959	0,9957	0,9959
		<b>0,9511</b>	<b>0,8742</b>	<b>0,8728</b>	<b>0,8748</b>	<b>0,8766</b>	<b>0,8760</b>	<b>0,9639</b>	<b>0,9607</b>	<b>0,9630</b>	<b>0,9636</b>	<b>0,9644</b>
Average		0,9511	0,8742	0,8728	0,8748	0,8766	0,876	0,9639	0,9607	0,963	0,9636	0,9644



• AUC

AUC		J48		J48Consolidated				Bagging				
		defecto	defecto	Bootstrap				defecto	J48			
			Balanced + Cov=99%	N_S=10	N_S=30	N_S=50	N_S=100	REPTre + N_S=10	N_S=10	N_S=30	N_S=50	N_S=100
Datasets	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	
abalone19	(50)	0,5	0,6616	0,6399	0,6294	0,6589	0,6502	0,7587	0,6826	0,7356	0,7541	0,7728
abalone9-18	(50)	0,6599	0,7147	0,7034	0,707	0,7106	0,7155	0,8201	0,8156	0,8524	0,8629	0,8692
ecoli-0_vs-1	(50)	0,9829	0,9829	0,9829	0,9829	0,9829	0,9829	0,9869	0,9824	0,9822	0,9821	0,9817
ecoli1	(50)	0,9155	0,9137	0,902	0,9037	0,9121	0,9133	0,9532	0,9498	0,9531	0,9531	0,9543
ecoli2	(50)	0,8461	0,8694	0,8634	0,8621	0,8679	0,8652	0,9345	0,9412	0,9502	0,9508	0,9488
ecoli3	(50)	0,7924	0,8807	0,8795	0,8778	0,8882	0,886	0,9207	0,9202	0,9337	0,9334	0,9349
ecoli4	(50)	0,7991	0,889	0,8815	0,8807	0,8819	0,89	0,9117	0,9278	0,9398	0,9534	0,97
glass-0-1-2-3_vs-4-5-6	(50)	0,8738	0,9022	0,8968	0,8941	0,8973	0,8871	0,9642	0,9696	0,9746	0,9748	0,9748
glass0	(50)	0,7967	0,8126	0,8132	0,8249	0,8171	0,8117	0,9022	0,9043	0,9121	0,9124	0,915
glass1	(50)	0,7093	0,7191	0,7076	0,7213	0,7153	0,7219	0,834	0,842	0,8555	0,8593	0,8617
glass2	(50)	0,69	0,7535	0,7331	0,7324	0,7494	0,7468	0,7735	0,8176	0,8446	0,8448	0,8471
glass4	(50)	0,7835	0,874	0,8756	0,8771	0,8796	0,8752	0,9376	0,9505	0,9744	0,9742	0,9748
glass5	(50)	0,99	0,9463	0,9402	0,9456	0,9478	0,9463	0,9718	0,9926	0,9928	0,9933	0,9927
glass6	(50)	0,8963	0,931	0,9245	0,931	0,931	0,9305	0,9414	0,9451	0,9514	0,9518	0,952
haberman1mb	(50)	0,5679	0,6315	0,629	0,606	0,6099	0,6067	0,6704	0,6597	0,6737	0,6837	0,6927
iris0	(50)	0,984	0,984	0,984	0,984	0,984	0,984	1	0,984	0,984	0,984	0,984
new-thyroid1	(50)	0,9217	0,9565	0,9527	0,96	0,9525	0,9505	0,9804	0,9868	0,9927	0,9934	0,9949
new-thyroid2	(50)	0,9269	0,9567	0,9449	0,9603	0,9588	0,959	0,9903	0,9936	0,996	0,996	0,9959
page-blocks0	(50)	0,9388	0,9658	0,9598	0,9654	0,9669	0,9655	0,9893	0,9884	0,9904	0,9906	0,9906
pima1mb	(50)	0,7448	0,7282	0,7245	0,7245	0,7265	0,7232	0,821	0,8053	0,8199	0,8237	0,8269
segment0	(50)	0,9838	0,987	0,987	0,987	0,9874	0,9876	0,9955	0,9957	0,9972	0,9975	0,9977
vehicle0	(50)	0,9398	0,9516	0,9516	0,9509	0,954	0,9551	0,9858	0,9885	0,9918	0,9922	0,9926
vehicle0	(50)	0,7083	0,7534	0,7568	0,7505	0,7581	0,7512	0,8305	0,8327	0,8513	0,8532	0,8566
vehicle0	(50)	0,9448	0,9532	0,952	0,9551	0,956	0,9546	0,9904	0,9917	0,995	0,9953	0,9966
vehicle0	(50)	0,7416	0,7795	0,7808	0,7687	0,7655	0,7811	0,8368	0,8414	0,8589	0,8619	0,8644
vowel0	(50)	0,9643	0,9629	0,9604	0,9628	0,9626	0,962	0,9891	0,986	0,9935	0,9957	0,9968
wisconsin1mb	(50)	0,9508	0,9513	0,9527	0,9503	0,9518	0,9496	0,9898	0,9898	0,9921	0,9923	0,9926
yeast-2_vs-8	(50)	0,5048	0,7801	0,7957	0,7845	0,7855	0,7779	0,7804	0,7815	0,8522	0,8682	0,8757
yeast1	(50)	0,7157	0,7211	0,715	0,7164	0,7153	0,7171	0,7877	0,7835	0,7974	0,8004	0,8023
yeast3	(50)	0,8951	0,9306	0,9334	0,9313	0,933	0,9318	0,9676	0,9604	0,9698	0,9727	0,9729
yeast4	(50)	0,7496	0,8662	0,863	0,8683	0,8701	0,8686	0,9106	0,897	0,9209	0,927	0,9293
yeast5	(50)	0,9083	0,9592	0,9576	0,9577	0,9567	0,9599	0,9794	0,9761	0,9809	0,9809	0,9819
yeast6	(50)	0,7527	0,8426	0,8463	0,8491	0,8517	0,8425	0,8837	0,8606	0,8663	0,8772	0,878
		0,8206	0,8640	0,8603	0,8607	0,8632	0,8621	0,9088	0,9074	0,9205	0,9238	0,9264
<b>Average</b>		<b>0,8206</b>	<b>0,864</b>	<b>0,8603</b>	<b>0,8607</b>	<b>0,8632</b>	<b>0,8621</b>	<b>0,9088</b>	<b>0,9074</b>	<b>0,9205</b>	<b>0,9238</b>	<b>0,9264</b>



• Kappa

Kapa		J48		J48Consolidated				Bagging				
		defecto	defecto	Bootstrap				defecto	J48			
			Balanced + Cov=99%	N_S=10	N_S=30	N_S=50	N_S=100	REPtree + N_S=10	N_S=10	N_S=30	N_S=50	N_S=100
Datasets	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	
abalone19	(50)	0	0,0282	0,0238	0,0254	0,0262	0,0252	0	0	0	0	0
abalone9-18	(50)	0,3282	0,1735	0,1756	0,1808	0,1716	0,173	0,2938	0,3905	0,4023	0,4283	0,4075
ecoli-0_vs-1	(50)	0,9679	0,9679	0,9679	0,9679	0,9679	0,9679	0,9724	0,9669	0,9649	0,9659	0,9659
ecoli1	(50)	0,7001	0,7088	0,6932	0,7149	0,7258	0,7249	0,7234	0,7149	0,7252	0,7251	0,7119
ecoli2	(50)	0,7494	0,7159	0,7043	0,7095	0,715	0,7225	0,759	0,768	0,7746	0,7825	0,7856
ecoli3	(50)	0,5465	0,5322	0,5285	0,5142	0,5313	0,544	0,5476	0,5644	0,5897	0,5779	0,5899
ecoli4	(50)	0,6565	0,5319	0,5015	0,5235	0,5115	0,5294	0,716	0,7438	0,7387	0,7359	0,7376
glass-0-1-2-3_vs-4-5-6	(50)	0,7983	0,7375	0,7414	0,7358	0,7405	0,7306	0,8054	0,8026	0,8103	0,8035	0,808
glass0	(50)	0,5684	0,5677	0,5618	0,5665	0,5574	0,5624	0,6355	0,6331	0,6538	0,6571	0,6695
glass1	(50)	0,4313	0,4076	0,4095	0,41	0,4315	0,431	0,5226	0,5056	0,519	0,5255	0,5532
glass2	(50)	0,1965	0,2345	0,212	0,1998	0,2304	0,2175	0,0136	0,0562	0,0211	0,0507	0,0619
glass4	(50)	0,5234	0,4687	0,464	0,455	0,4788	0,4784	0,283	0,5385	0,497	0,5114	0,5497
glass5	(50)	0,8303	0,4501	0,4169	0,4586	0,4708	0,4501	0,252	0,693	0,731	0,7152	0,6998
glass6	(50)	0,7835	0,7487	0,7541	0,7462	0,7448	0,7453	0,8363	0,7969	0,8198	0,8169	0,8229
haberman1mb	(50)	0,1474	0,2344	0,2362	0,2112	0,215	0,2178	0,185	0,15	0,1516	0,1534	0,1564
iris0	(50)	0,9749	0,9749	0,9749	0,9749	0,9749	0,9749	1	0,9749	0,9749	0,9749	0,9749
new-thyroid1	(50)	0,8602	0,8841	0,8746	0,8927	0,8745	0,8767	0,8682	0,8982	0,9016	0,8991	0,8965
new-thyroid2	(50)	0,8641	0,8847	0,8561	0,8863	0,8898	0,8886	0,8679	0,877	0,8909	0,8891	0,8928
page-blocks0	(50)	0,8414	0,7788	0,7699	0,7765	0,7776	0,7791	0,8583	0,8609	0,8655	0,8663	0,8659
pima1mb	(50)	0,4026	0,4118	0,4031	0,4045	0,4091	0,4022	0,4562	0,4313	0,4562	0,4644	0,4665
segment0	(50)	0,9669	0,9492	0,9498	0,9502	0,9497	0,95	0,976	0,9718	0,9713	0,9714	0,972
vehicle0	(50)	0,8376	0,8046	0,805	0,806	0,8073	0,8083	0,8609	0,8643	0,8767	0,8853	0,8857
vehicle0	(50)	0,345	0,4	0,4096	0,4152	0,4248	0,4188	0,3417	0,3832	0,3945	0,3924	0,3934
vehicle0	(50)	0,8906	0,8594	0,8583	0,8695	0,874	0,8692	0,9187	0,9366	0,9392	0,9401	0,9417
vehicle0	(50)	0,3454	0,4268	0,4258	0,4188	0,4097	0,425	0,351	0,3848	0,3798	0,3762	0,372
vowel0	(50)	0,9327	0,7672	0,759	0,7602	0,7643	0,7554	0,9092	0,9226	0,9368	0,9374	0,9379
wisconsin1mb	(50)	0,8883	0,8909	0,8884	0,8878	0,8897	0,8844	0,9147	0,9166	0,921	0,9217	0,9224
yeast-2_vs-8	(50)	0,025	0,3045	0,3007	0,2942	0,2913	0,2907	0,6002	0,0884	0,0386	0,0231	0,0156
yeast1	(50)	0,3536	0,3662	0,3655	0,3634	0,365	0,3687	0,3864	0,3942	0,4024	0,4092	0,4126
yeast3	(50)	0,7334	0,7346	0,7369	0,7344	0,734	0,734	0,7475	0,7389	0,7385	0,7374	0,7347
yeast4	(50)	0,3136	0,1765	0,1793	0,1741	0,1778	0,1812	0,259	0,3278	0,3583	0,3614	0,3687
yeast5	(50)	0,7559	0,5194	0,5003	0,5194	0,5085	0,5296	0,7321	0,7434	0,7568	0,7757	0,7748
yeast6	(50)	0,4808	0,2416	0,2537	0,253	0,2502	0,2497	0,4883	0,5631	0,5643	0,5665	0,56
		0,6073	0,5722	0,5667	0,5697	0,5724	0,5729	0,6085	0,6243	0,6293	0,6315	0,6336
<b>Average</b>		<b>0,6073</b>	<b>0,5722</b>	<b>0,5667</b>	<b>0,5697</b>	<b>0,5724</b>	<b>0,5729</b>	<b>0,6085</b>	<b>0,6243</b>	<b>0,6293</b>	<b>0,6315</b>	<b>0,6336</b>





# SEGUNDA FASE

## ● Accuracy

ACCURACY	Porcentajes de Consolidación	PCTBagging																																				
		Balanceadas y 99% coverage										100 muestras Bootstrap																										
		0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %															
Datasets		1	2	3	4	5	6	7	8	9	10	11																										
abalone19	(50)	81.9144	81.4256	81.1669	80.8291	80.8364	80.8555	80.8076	81.0423	81.4951	82.0103	82.5517	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334		
abalone9-18	(50)	79.0425	78.5907	78.0158	76.9748	77.0845	76.3867	76.4141	76.5506	76.5096	76.4963	76.6195	95.5134	95.3357	95.1444	95.0896	94.6383	94.6383	94.5425	94.4875	94.4054	94.3643	94.3644															
ecoli-0_vs-1	(50)	98.9091	98.9091	98.9091	98.9091	98.9091	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545															
ecoli1	(50)	87.259	86.9908	86.9315	87.3481	87.3486	87.5869	87.8837	88.2401	88.2704	88.36	88.4495	90.4473	90.1493	90.0277	89.6106	89.7291	89.7296	89.9069	89.788	89.7577	89.8468	90.9056															
ecoli2	(50)	90.953	90.9829	90.4473	90.5369	90.507	90.7744	90.7744	90.8639	90.8639	92.2634	92.2634	94.7643	94.4952	94.1989	94.0198	93.8701	93.6317	93.572	93.4245	93.514	93.3345	93.245															
ecoli3	(50)	85.1752	85.1752	85.1444	85.1743	85.3235	85.6523	85.6523	85.5926	86.694	86.9333	86.9333	93.3955	93.5737	93.0707	93.1585	93.0092	92.741	92.4728	92.2046	92.1738	92.1747	92.1159															
ecoli4	(50)	90.8604	90.8604	88.8389	88.6905	88.601	88.0057	88.0057	88.0953	88.8393	91.4008	91.4008	97.2629	97.2331	97.1137	96.9653	96.8464	96.6383	96.6383	96.6383	96.608	96.6664	96.6664															
glass-0-1-2-3_vs-4-5	(50)	90.1495	90.1495	90.0554	90.0089	89.9612	89.7741	89.7741	90.0066	89.9136	89.8206	89.8206	93.5482	93.4075	93.4064	93.5493	93.5039	92.897	92.9435	92.7542	92.6111	92.5692	92.3832															
glass0	(50)	79.6135	79.0975	78.588	78.5869	78.9192	79.1052	79.3378	79.3378	79.2935	79.4341	79.6235	85.5615	84.5725	83.4961	82.3754	82.1473	81.4496	81.2159	81.0786	81.0332	81.4097	81.7818															
glass1	(50)	73.6844	73.1196	73.2636	72.6578	71.8106	71.1584	70.9701	71.5869	71.3477	71.6733	72.0011	79.8981	78.9258	77.5648	76.4972	75.5161	75.4695	75.423	74.9576	74.7708	74.722	75.1417															
glass2	(50)	69.7065	69.5648	68.3101	68.5426	68.9169	68.6866	68.8283	72.7597	73.0454	74.8217	74.8217	91.8261	91.6855	91.4042	90.9856	90.66	90.381	90.4252	90.2392	90.2392	90.2381	90.0986															
glass4	(50)	86.6833	86.6833	86.7298	86.9192	86.9192	87.1517	87.1517	87.1517	87.1517	89.7685	89.8616	89.8616	95.7099	94.7763	94.5426	94.3555	94.2148	94.1207	94.1207	94.3555	94.3555	94.5858	94.4928														
glass5	(50)	87.9911	87.9911	87.9911	88.0388	88.0388	88.2248	88.2248	88.2248	88.2248	89.7475	89.7475	97.7608	97.7608	97.7619	97.7619	98.041	98.7386	98.7386	98.6921	98.6921	98.5537	98.5837															
glass6	(50)	93.6412	93.6412	93.5482	93.5017	93.5017	93.175	93.175	93.175	92.8472	93.1728	93.1728	95.7453	95.6988	95.9336	95.887	95.7918	95.5592	95.5592	95.2315	95.0451	95.0421																
haberman	(50)	68.1703	68.8911	67.8424	67.0254	66.3707	66.0841	65.7911	66.2152	65.66	65.8567	66.184	71.9947	71.5357	71.6039	71.4389	71.3406	71.339	71.3406	71.3728	71.5029	71.4373	71.4701															
iris0	(50)	100	100	100	100	100	100	98.9333	98.9333	98.9333	98.9333	100	100	100	100	100	100	98.9333	98.9333	98.9333	98.9333	98.9333	98.9333															
new-thyroid1	(50)	96.6047	96.6047	96.6047	96	96	95.9535	95.9535	95.9535	97.0233	97.0233	97.0233	97.5349	97.3023	96.2326	95.9535	95.9535	95.8605	95.8605	95.9635	95.907	95.814	95.8605															
new-thyroid2	(50)	96.7907	96.7907	96.6512	96.5116	96.5116	96.186	96.186	96.186	96.6512	96.7442	96.7442	97.5349	97.0698	96.3256	96.0465	95.907	96.0465	96.0465	96.0465	96.0465	96.1395	96.1395															
page-blocks0	(50)	93.3033	95.243	95.1791	95.0091	94.9653	94.9617	95.0384	95.0877	95.0859	95.1973	95.2851	97.5987	97.5274	97.4324	97.3027	97.2351	97.151	97.151	97.6372	97.1071	97.0633	97.0359	97.0359														
svm	(50)	73.1636	72.465	72.4344	72.1353	71.6933	71.5364	71.2371	71.2113	71.3803	71.7187	76.1316	75.9108	73.9274	73.3201	73.0863	72.5786	72.6312	72.4629	72.4751	72.6573	72.6794																
segment0	(50)	98.9686	98.8387	98.8387	98.8474	98.8387	98.8474	98.8474	98.8474	98.8474	98.7868	98.7868	98.7521	98.7348	99.3412	99.2763	99.2633	99.1983	99.2069	99.2112	99.1809	99.1548	99.1722	99.2069														
vehicle0	(50)	93.1806	93.1804	93.1804	93.1921	93.0616	92.7431	92.6601	92.5661	92.5661	92.8254	95.8273	95.6852	95.3664	94.8702	94.6223	94.3266	94.1609	96.1209	96.1209	96.1209	96.1209	96.1209															
vehicle0	(50)	74.8814	73.7342	73.5809	73.6774	73.3202	73.1198	72.8361	72.7888	72.9426	73.0255	73.0844	78.6873	77.9534	77.0082	76.2171	75.7556	75.8037	75.8037	75.5439	75.2177	75.1534	75.3067															
vehicle0	(50)	96.1113	95.8278	95.4727	94.9396	94.7978	94.7268	94.6677	94.6204	94.6913	94.6734	94.6675	97.6118	97.6118	96.9652	96.3231	96.1696	96.1814	96.1932	96.1344	96.1344	96.1344	96.1344															
vehicle0	(50)	75.3673	74.1966	74.0553	73.8545	74.1144	74.5044	74.5039	74.5274	74.5159	74.6812	74.6812	99.0077	97.8138	97.3398	97.3755	96.8677	96.8908	96.5602	96.3584	96.3114	96.2119	96.2324															
vowel0	(50)	95.5065	95.5258	95.3535	95.2724	95.2217	95.0295	95.0295	95.2119	95.1308	95.3332	95.3332	99.0686	99.0686	99.0686	99.0686	98.3804	98.3805	98.3704	98.3608	98.4211	98.4515	98.5222															
wisconsinmb	(50)	95.7101	95.4469	95.4325	95.2862	95.257	95.2131	95.1838	95.1838	95.1397	95.0372	95.0157	96.1712	96.1787	96.1197	95.9731	95.651	95.4609	95.4021	95.3143	95.2119	95.1086	95.1529															
yeast-2_vs-8	(50)	86.4957	86.5163	86.4957	85.2706	84.7081	84.2017	84.2302	84.5423	84.7665	85.1821	85.244	95.8915	95.8707	95.767	95.8913	95.8913	96.4725	96.4725	96.4936	96.5767	96.4938	96.4938															
yeast1	(50)	72.2309	72.2035	70.9306	70.5729	70.1552	70.4383	70.492	70.6268	70.674	70.8625	71.0714	77.6554	76.6111	75.641	75.014	74.603	74.165	73.9697	74.0371	73.9999	74.1111	74.1785															
yeast3	(50)	93.5917	93.4905	93.3827	93.4908	93.5178	93.6256	93.639	93.8411	93.8815	93.9849	93.9849	94.9122	94.9594	94.9661	94.7303	94.6428	94.6496	94.5485	94.5216	94.5216	94.5486	94.5756															
yeast4	(50)	79.1485	79.1373	78.7534	78.5512	78.4231	78.2949	78.3959	78.9348	79.0495	79.8787	79.9931	96.9946	96.9879	96.779	96.6846	96.543	96.4555	96.4419	96.4419	96.4554	96.4891	96.4898															
yeast5	(50)	94.0026	94.0026	93.9285	92.8703	92.8703	92.7964	92.7964	92.7964	94.8315	95.0136	98.713	98.6455	98.5714	98.4703	98.3895	98.4098	98.3895	98.3761	98.3963	98.3761	98.3963	98.4958															
yeast6	(50)	88.5377	88.5377	88.5377	88.0862	88.0862	86.7384	86.7384	86.7384	87.466	87.466	87.466	98.2615	98.248	98.2615	98.2547	98.2075	98.2277	98.1873	98.1335	98.0728	98.0796	98.0661															
		86.9510	86.7833	86.5029	86.2728	86.1997	86.0236	86.0211	86.2402	86.5537	86.8498	86.9294	92.6858	92.4009	92.0618	91.8195	91.6588	91.5240	91.4854	91.4184	91.3824	91.3788	91.4052															
Average		86.951	86.7833	86.5029	86.2728	86.1997	86.0236	86.0211	86.2402	86.5537	86.8498	86.9294	92.6858	92.4009	92.0618	91.8195	91.6588	91.5240	91.4854	91.4184	91.3824	91.3788	91.4052															

PCT	50% Atributos																					
	Balanceadas y 99% coverage										100 muestras modo RSM											
	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Datasets	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
80.0241	78.1171	77.7648	78.4982	79.0828	79.4374	79.8421	80.532	81.0782	81.8713	82.5947	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334
80.0554	80.9981	81.1357	79.8915	79.7416	78.4545	78.3449	78.7409	78.6863	78.8649	79.1111	94.8432	95.1032	95.0345	94.8292	94.7333	94.5424	94.6104	94.5009	94.4873	94.542	94.5693	
98.3636	98.3182	98.3182	98.3636	98.4545	98.0455	98.0455	97.8636	97.8636	98	98	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.5455	98.5455	98.5455	98.5455	98.5455	98.5455
86.8442	86.5162	86.4271	86.7252	86.7252	86.4545	86.4545	87.1422	87.4109	87.5	87.4996	88.9877	89.1352	89.3126	89.3714	89.2818	88.9249	88.9249	88.9846	88.9846	89		

RSM

75% Atributos

Table with 20 columns (0.00% to 100.00%) and 20 rows of data. Includes sub-headers 'Balanceadas y 99% coverage' and '100 muestras modo RSM'. The table contains numerical data for various samples.

PC1

50% Atributos

Table with 20 columns (0.00% to 100.00%) and 20 rows of data. Includes sub-headers 'Balanceadas y 99% coverage' and '100 muestras modo RSM'. The table contains numerical data for various samples.

TRF

75% Atributos																						
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	
Balancedas y 99% coverage										100 muestras mo RSM												
78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	99
81.5526	80.8243	80.3403	80.3357	80.3141	80.3285	80.422	80.822	81.0041	81.524	82.2044	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334	99.2334
79.5899	79.3842	79.2334	78.5764	78.6871	77.7266	77.7403	77.3999	77.3724	77.6865	77.7687	94.9794	94.7882	94.7472	94.6378	94.583	94.5557	94.5556	94.5145	94.501	94.5421	94.5557	94.5557
98.9091	98.9091	98.9091	98.9091	98.9091	98.4545	98.4545	98.4545	98.4545	98.4545	98.4545	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273	98.7273
86.7239	86.3665	86.3064	87.0808	87.0821	87.3512	87.4706	87.9157	87.7985	88.0364	88.1558	89.8503	89.5808	89.4917	89.5821	89.3156	89.3152	89.3152	89.3152	89.4921	89.7608	89.7608	89.7608
90.6558	90.3003	89.7673	89.7059	89.6462	90.5083	90.5083	90.6879	90.4789	92.2041	92.2331	94.2884	94.1984	94.0492	93.8705	93.9306	93.8112	93.7818	93.752	93.8112	93.7217	93.7217	93.7217
84.3411	84.3411	84.0123	83.9829	84.6997	85.2963	85.2963	85.1462	86.604	86.6927	86.6927	93.036	92.856	92.7388	92.2015	92.1721	92.5299	92.5	92.3815	92.4991	92.4403	92.3815	92.3815
91.4583	91.4583	90.2103	90.0917	89.8529	87.8551	87.8551	88.0342	88.5992	91.8459	91.8459	96.9653	96.9653	96.8165	96.6383	96.6681	96.5487	96.5487	96.5487	96.5487	96.5487	96.5487	96.5487
90.5227	90.2425	90.0078	90.1019	89.7752	89.68	89.6334	89.773	89.959	90.0997	90.1462	93.0797	92.9867	93.0808	93.1274	93.0343	92.7087	92.7087	92.7087	92.7087	92.7087	92.7087	92.7087
79.7154	79.5227	78.4507	78.7774	79.4363	79.6667	79.4795	79.3854	78.8726	79.2458	79.5227	81.7663	81.0221	80.8804	80.5072	80.2735	80.0399	80.0399	80.0864	80.0864	80.0864	80.0864	80.0864
74.433	72.4286	73.124	72.6512	71.4319	71.3887	71.7209	70.7409	70.7818	71.0199	71.3466	77.237	76.9048	76.3433	75.353	75.4983	75.2647	75.3134	75.268	75.1761	75.1772	75.0831	75.0831
68.6312	68.1152	66.4839	66.206	66.6346	67.1528	67.1993	72.8117	72.9989	74.9535	75.0465	91.1251	91.124	91.0753	91.1694	91.2182	91.0753	91.1218	91.0288	91.0753	91.0753	90.9823	90.9823
87.2935	87.2935	87.2935	86.5404	86.5404	86.8195	86.8195	86.8195	86.8195	86.8195	86.8195	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744	89.6744
87.9446	87.9446	87.9446	87.9911	87.9911	88.2248	88.2248	88.2248	88.2248	89.557	89.557	89.557	98.6921	98.6921	98.6921	98.6921	98.6921	98.6921	98.6921	98.6921	98.6921	98.6921	98.6921
94.062	94.062	93.6423	93.5028	93.5028	93.1296	93.1296	93.1296	92.8472	92.1293	92.1293	95.0377	95.1318	95.1329	95.0399	95.0875	95.041	95.041	95.041	94.948	94.948	94.948	94.948
67.9466	69.1883	68.568	66.9313	66.6658	66.734	67.0275	66.899	67.065	67.3273	67.2956	72.6198	72.6198	72.2285	72.2263	72.1608	71.1137	71.1137	71.1137	71.1793	70.982	70.982	70.982
100	100	100	100	100	98.9333	98.9333	98.9333	98.9333	98.9333	98.9333	100	100	100	100	100	98.9333	98.9333	98.9333	98.9333	98.9333	98.9333	98.9333
96.4186	96.4186	96.3256	95.814	95.814	95.2093	95.2093	95.2093	96.4186	96.4186	96.4186	96.2791	96.4651	96.3256	96.1395	96.093	96.093	96.093	96.093	96.0465	96.0465	96.0465	96.0465
96.3256	96.3256	96.3256	95.3488	95.3488	94.9767	94.9767	94.9767	96.2326	96.093	96.093	96.6512	96.7442	96.5581	96.5581	96.5116	96.5581	96.5581	96.4651	96.4186	96.4186	96.4186	96.4186
95.1681	95.1846	95.159	94.9178	94.8831	94.8885	94.9561	95.0456	95.1114	95.2357	95.3344	97.5018	97.3191	97.2496	97.1838	97.1601	97.1491	97.14	97.1436	97.1308	97.1235	97.1253	97.1253
73.1361	72.852	72.5408	72.8001	71.5221	71.6793	71.7173	71.7316	71.7309	71.796	71.8478	74.8836	74.0245	73.906	73.4758	73.202	73.0983	73.2024	73.2155	73.2547	73.2154	73.2153	73.2153
99.051	98.999	98.934	98.895	98.8647	98.83	98.7824	98.7564	98.7694	98.778	98.7911	99.3065	99.2459	99.2199	99.2069	99.1982	99.1982	99.1982	99.1982	99.1939	99.1939	99.1939	99.1939
93.5109	93.4398	93.1211	92.9916	92.9323	92.8141	92.9318	92.8493	92.8845	92.9199	92.9317	95.2962	94.8269	94.6933	94.3502	94.1848	94.0547	94.0901	94.1376	94.1376	94.161	94.1847	94.1847
74.4558	73.9238	73.3217	73.2509	73.2615	73.2739	73.3215	73.3334	73.1563	73.0854	73.5227	78.5227	77.5999	77.0993	76.6898	76.5835	76.2761	76.2995	76.1694	76.2047	76.2284	76.2403	76.2403
96.3234	95.8036	95.8863	95.0114	95.2719	95.0467	94.8457	94.9165	94.8811	94.8692	96.8791	96.4533	96.2999	96.2999	96.0395	95.989	95.8034	95.7799	95.8154	95.8036	95.7799	95.7799	95.7799
75.3683	73.7359	73.5951	73.7717	73.6893	73.9963	74.0083	74.0794	74.1621	74.2455	74.4704	77.9805	77.117	76.5855	76.4793	76.3372	76.2667	76.0657	76.16	76.0776	76.0658	76.0775	76.0775
95.7885	95.7276	95.6266	95.3836	95.09	94.999	94.989	95.141	95.1813	95.5458	95.5761	98.9472	98.9898	98.9878	98.9371	98.8763	98.8864	98.8864	98.8661	98.8965	98.8965	98.8965	98.8965
95.7691	95.5794	95.6236	95.5639	95.4323	95.228	95.3448	95.4329	95.2572	95.3453	95.4332	95.5351	95.1985	95.0814	95.0966	95.1111	94.9355	95.023	94.9791	94.9644	94.935	94.981	94.981
88.2171	88.3838	88.4051	86.765	85.2927	83.9214	84.1028	84.1293	84.2307	85.064	85.3954	95.6841	95.6841	95.5595	95.5801	95.5801	95.6632	95.6632	95.6632	95.6632	95.6632	95.6632	95.6632
72.8305	72.4732	71.631	70.9438	70.9639	70.7751	71.1462	71.3079	71.2202	71.4292	71.4828	76.4016	75.9568	75.1665	75.5257	75.364	75.2495	75.1146	75.108	75.108	75.0743	75.0743	75.0743
93.4772	93.3493	93.1404	93.2819	93.2887	93.2887	93.3021	93.794	93.8614	93.8681	93.8185	95.101	94.9999	94.9796	94.8718	94.7506	94.7303	94.7506	94.7506	94.7033	94.6764	94.6966	94.6966
80.2495	80.5326	80.762	80.0609	79.2117	78.3891	78.4969	78.7189	78.7594	78.9826	79.8309	96.6846	96.6375	96.6509	96.6307	96.57	96.5498	96.5498	96.5296	96.5565	96.5363	96.5296	96.5363
94.6158	94.6158	94.5821	92.6546	92.6546	92.1966	92.1966	92.1966	94.5824	94.6295	94.6295	98.5781	98.6118	98.5377	98.5445	98.5511	98.5377	98.5377	98.5444	98.5511	98.5377	98.5309	98.5309
89.5351	89.5351	89.1512	88.727	88.9896	88.2617	88.2617	88.228	88.888	89.1236	89.1236	98.0593	98.0728	98.0863	98.0054	98.0054	97.965	97.9718	97.9583	97.9447	97.9313	97.9583	97.9583
87.0915	86.8867	86.6161	86.2899	86.1721	85.9704	86.0154	86.2107	86.5565	86.9306	87.0249	92.1208	91.9391	91.8234	91.6953	91.6250	91.4979	91.4970	91.4877	91.4857	91.4795	91.4737	91.4737
87.0915	86.8867	86.6161	86.2899	86.1721	85.9704	86.0154	86.2107	86.5565	86.9306	87.0249	92.1208	91.9391	91.8234	91.6953	91.625	91.4979	91.497	91.4877	91.4857	91.4795	91.4737	91.4737

● **Balanced Accuracy**

BALANCED ACCURACY	PCTBagging																						
	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	
	Balancedas y 99% coverage										100 muestras Bootstrap												
Datasets	1	2	3	4	5	6	7	8	9	10	11												
abalone19	(50)	0.6844	0.6881	0.6738	0.67375	0.67075	0.66255	0.6609	0.66635	0.66245	0.6618	0.65645	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
abalone9-18	(50)	0.7478	0.7503	0.7441	0.73875	0.74025	0.73785	0.7368	0.73635	0.7372	0.735	0.73565	0.63335	0.65915	0.66165	0.66475	0.667	0.65885	0.6583	0.6604	0.6576	0.65515	0.654
ecoli-0	(50)	0.9857	0.9857	0.9857	0.9857	0.9822	0.9822	0.9822	0.9822	0.9822	0.9819	0.9819	0.9819	0.9819	0.9819	0.9819	0.9829	0.9829	0.9829	0.9829	0.9829	0.9829	0.9829
ecoli1	(50)	0.90015	0.898	0.898	0.89755	0.89755	0.8942	0.8953	0.89575	0.8951	0.89435	0.8945	0.85605	0.85415	0.8477	0.8496	0.8555	0.85445	0.85925	0.8579	0.8563	0.85835	0.85725
ecoli2	(50)	0.88985	0.89005	0.88365	0.88415	0.884	0.8802	0.8802	0.8791	0.87825	0.87895	0.87895	0.87165	0.8732	0.8692	0.8626	0.86015	0.8575	0.84485	0.84715	0.8516	0.84655	0.84365
ecoli3	(50)	0.8756	0.8756	0.87795	0.8819	0.88145	0.87825	0.8741	0.874	0.87405	0.87405	0.87615	0.87605	0.877	0.87125	0.8779	0.8772	0.8761	0.87635	0.87619	0.876		

50% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balancedas y 99% coverage											100 muestras modo RSM										
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
0,7002	0,69675	0,6749	0,6689	0,6659	0,6528	0,6504	0,6484	0,6472	0,63955	0,63685	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,7421	0,7325	0,73205	0,7325	0,72965	0,72295	0,72235	0,72095	0,7195	0,7203	0,72045	0,56245	0,59855	0,6026	0,6099	0,6141	0,6208	0,62665	0,62825	0,62935	0,63185	0,6332
0,97705	0,9767	0,9767	0,9773	0,9786	0,9758	0,9758	0,97465	0,97465	0,97695	0,97695	0,9819	0,9819	0,9819	0,9819	0,9819	0,9829	0,9829	0,9829	0,9829	0,9829	0,9829
0,8935	0,891	0,89085	0,88955	0,8886	0,88725	0,88775	0,89035	0,893	0,8922	0,89175	0,8426	0,8577	0,85325	0,8489	0,8452	0,84805	0,8494	0,85205	0,8432	0,84365	0,84365
0,8858	0,8851	0,8762	0,87415	0,87675	0,87495	0,87495	0,87325	0,87395	0,88	0,88	0,8533	0,8601	0,86535	0,8567	0,85745	0,85005	0,85005	0,85825	0,86115	0,85965	0,8598
0,87445	0,87445	0,8662	0,8665	0,86665	0,8718	0,8718	0,8702	0,8734	0,86525	0,86525	0,69255	0,70915	0,72685	0,74225	0,755	0,75995	0,75995	0,75755	0,75335	0,75305	0,75255
0,90695	0,90695	0,9003	0,89855	0,89475	0,8591	0,8591	0,86305	0,86605	0,8838	0,8838	0,7581	0,7581	0,80215	0,81885	0,821	0,8179	0,8179	0,8179	0,8179	0,80835	0,80835
0,93175	0,9268	0,9172	0,91645	0,91455	0,90845	0,90895	0,9072	0,9062	0,9064	0,9067	0,9034	0,914	0,9169	0,9147	0,90835	0,89585	0,89325	0,89225	0,89155	0,89155	0,89335
0,82285	0,81535	0,8114	0,81185	0,8137	0,8176	0,81585	0,81055	0,80665	0,80805	0,81195	0,82695	0,8179	0,81415	0,8082	0,79985	0,79665	0,7935	0,79175	0,7876	0,79185	0,7933
0,7476	0,7122	0,71785	0,7241	0,72265	0,72185	0,7228	0,72135	0,7249	0,72025	0,718	0,7504	0,73435	0,7221	0,718	0,71625	0,7142	0,71165	0,7102	0,7095	0,70755	0,7083
0,7547	0,74825	0,7293	0,7228	0,7198	0,7055	0,7055	0,737	0,7408	0,7362	0,7362	0,4982	0,49995	0,5158	0,5343	0,5666	0,60265	0,6099	0,6094	0,60915	0,6069	0,6005
0,90415	0,90415	0,9034	0,8735	0,8735	0,86775	0,86775	0,86775	0,87	0,87125	0,87125	0,7305	0,7431	0,7994	0,8039	0,79665	0,79615	0,7964	0,7987	0,7987	0,7852	0,7852
0,93295	0,93295	0,93295	0,93315	0,93315	0,93805	0,93805	0,93805	0,9439	0,9439	0,9439	0,74925	0,74925	0,88135	0,88135	0,88135	0,9411	0,9411	0,9411	0,9411	0,92185	0,92185
0,92405	0,92405	0,9235	0,9242	0,9222	0,9187	0,9187	0,9187	0,9133	0,9144	0,9144	0,8814	0,8851	0,8974	0,9085	0,9082	0,9048	0,9048	0,9045	0,89505	0,89505	0,89505
0,65275	0,6359	0,63995	0,64285	0,6402	0,63975	0,6385	0,63665	0,63795	0,6366	0,6371	0,5035	0,5043	0,52165	0,55395	0,55395	0,57125	0,57125	0,57125	0,57125	0,57125	0,5719
1	1	0,998	0,998	0,998	0,982	0,982	0,982	0,982	0,982	0,982	1	1	1	1	1	0,984	0,984	0,984	0,984	0,984	0,984
0,96875	0,96875	0,95865	0,9409	0,9409	0,94285	0,94285	0,94285	0,9539	0,9539	0,9539	0,931	0,935	0,92415	0,91875	0,9222	0,9248	0,9248	0,92195	0,9205	0,9194	0,9194
0,9759	0,9759	0,96745	0,93925	0,94005	0,93605	0,93605	0,9355	0,94815	0,94925	0,94925	0,9236	0,9293	0,92995	0,92285	0,92255	0,9277	0,9277	0,9254	0,9234	0,9246	0,9246
0,95605	0,95185	0,9518	0,95285	0,95185	0,94885	0,94945	0,9504	0,95005	0,94995	0,9503	0,90705	0,90985	0,91745	0,92205	0,92155	0,92085	0,9202	0,9196	0,91825	0,91775	0,91745
0,72375	0,71935	0,7201	0,71295	0,71065	0,70675	0,70635	0,7077	0,70725	0,7057	0,70665	0,67555	0,6997	0,69645	0,70235	0,6968	0,69605	0,69805	0,69675	0,69885	0,698	0,69875
0,98895	0,9885	0,98645	0,9859	0,98655	0,9857	0,9853	0,9853	0,98495	0,98475	0,98495	0,985	0,98285	0,9827	0,98235	0,9817	0,98145	0,9812	0,9812	0,9813	0,98125	0,98125
0,94985	0,94815	0,94405	0,9413	0,9383	0,93575	0,9345	0,9339	0,93305	0,93315	0,93245	0,95155	0,9494	0,9461	0,937	0,932	0,9276	0,92635	0,92355	0,92375	0,9226	0,92305
0,77555	0,76335	0,7627	0,7574	0,75415	0,7506	0,7463	0,74655	0,74155	0,74415	0,7421	0,65245	0,67085	0,67675	0,6757	0,6736	0,6725	0,67045	0,667	0,6661	0,66595	0,6679
0,97275	0,9658	0,9592	0,9524	0,95385	0,95445	0,9516	0,9479	0,94595	0,94445	0,94415	0,9748	0,96595	0,9631	0,95405	0,9486	0,94535	0,94345	0,94385	0,9441	0,94395	0,9436
0,77035	0,76245	0,7614	0,761	0,76005	0,7606	0,75785	0,75665	0,7565	0,75465	0,7541	0,63935	0,64205	0,6578	0,67135	0,67605	0,67575	0,67695	0,6781	0,67585	0,6747	0,67435
0,9744	0,9741	0,9688	0,9637	0,9603	0,9582	0,9585	0,95855	0,95815	0,9599	0,96	0,9688	0,9727	0,9727	0,97265	0,9703	0,96925	0,96815	0,965	0,96655	0,9665	0,9665
0,9688	0,96025	0,9599	0,9596	0,9594	0,9591	0,95885	0,95835	0,95745	0,95655	0,95658	0,96915	0,9581	0,9566	0,95545	0,9517	0,9488	0,9489	0,94705	0,94715	0,9465	0,9458
0,76935	0,77735	0,77525	0,7604	0,7587	0,74715	0,7503	0,75135	0,74815	0,7503	0,75115	0,5	0,5	0,49915	0,49915	0,49915	0,50675	0,50675	0,50675	0,50925	0,5086	0,5086
0,70725	0,70175	0,7056	0,7033	0,70105	0,69855	0,6997	0,699	0,6978	0,69825	0,6982	0,5587	0,66805	0,6756	0,666	0,66655	0,6663	0,666	0,6662	0,6666	0,6651	0,66505
0,9299	0,9272	0,9241	0,9206	0,9205	0,9228	0,9229	0,9246	0,92755	0,9287	0,9285	0,5274	0,8721	0,8917	0,89245	0,88035	0,8752	0,874	0,873	0,87295	0,87315	0,8738
0,83405	0,8291	0,816	0,8193	0,82725	0,81895	0,81825	0,8197	0,8204	0,821	0,8205	0,5	0,5047	0,5365	0,57145	0,60635	0,62785	0,6267	0,6304	0,6321	0,6312	0,6302
0,95405	0,95405	0,95235	0,9454	0,9439	0,94005	0,94005	0,94115	0,9493	0,9506	0,9506	0,5	0,64885	0,7789	0,8683	0,89205	0,8894	0,88515	0,8789	0,88445	0,8888	0,891
0,87905	0,8775	0,87705	0,8676	0,8767	0,86655	0,8667	0,8667	0,8582	0,8586	0,85905	0,5	0,52125	0,52975	0,5943	0,63225	0,7108	0,70935	0,72195	0,7189	0,7231	0,7248
0,8712	0,8669	0,8631	0,8590	0,8582	0,8539	0,8535	0,8541	0,8549	0,8552	0,8551	0,7486	0,7709	0,7859	0,7936	0,7964	0,8016	0,8014	0,8014	0,8008	0,7996	0,7997
0,8712	0,8669	0,8631	0,8590	0,8582	0,8539	0,8535	0,8541	0,8549	0,8552	0,8551	0,7486	0,7709	0,7859	0,7936	0,7964	0,8016	0,8014	0,8014	0,8008	0,7996	0,7997

3SM

75% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balancedas y 99% coverage											100 muestras modo RSM										
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
0,692	0,6882	0,6731	0,67145	0,66515	0,66215	0,6619	0,65565	0,6573	0,65535	0,65445	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,7422	0,73415	0,72955	0,7267	0,7276	0,72325	0,72355	0,72145	0,72085	0,7224	0,7233	0,62225	0,62765	0,62495	0,62435	0,62635	0,62635	0,6264	0,6262	0,62735	0,6297	0,63095
0,98155	0,98155	0,98155	0,9822	0,9822	0,978	0,978	0,97865	0,97865	0,9789	0,9789	0,9819	0,9819	0,9819	0,9819	0,9829	0,9829	0,9829	0,9829	0,9829	0,9829	0,9829
0,89925	0,8947	0,89255	0,89275	0,8883	0,88465	0,88625	0,8879	0,88965	0,89005	0,8899	0,86895	0,8625	0,86065	0,84435	0,84215	0,84455	0,846	0,84715	0,8425	0,84365	0,84365
0,88785	0,88695	0,8725	0,87275	0,8728	0,8781	0,8781	0,87635	0,8741	0,87915	0,87915	0,862	0,8623	0,8607	0,8551	0,8568	0,8531	0,8531	0,8587	0,8616	0,8598	0,8598
0,8795	0,8795	0,87805	0,8794	0,8732	0,8748	0,8748	0,8739	0,8696	0,85985	0,85985	0,719	0,72685	0,7391	0,74635	0,7482	0,7509	0,74935	0,74755	0,74635	0,74615	0,74615
0,8974	0,8974	0,87705	0,87565	0,87455	0,85895	0,85895	0,86305	0,8667	0,88095	0,88095	0,8051	0,8051	0,81495	0,8185	0,81835	0,8132	0,8132	0,8132	0,8132	0,80835	0,80835
0,92145	0,9211	0,9178	0,91225	0,9086	0,8985	0,8985	0,8994	0,90075	0,90565	0,90565	0,9049	0,90265	0,9025	0,9015	0,89845	0,89235	0,89125	0,89125	0,89125	0,89125	0,89305
0,8228	0,81875	0,8139	0,80985	0,81105	0,8116	0,8066	0,8062	0,8022	0,8015	0,80285	0,8146	0,80835	0,8056	0,7957	0,7964	0,79215	0,7911	0,79255	0,7933	0,7936	0,7936
0,73515	0,71645	0,718	0,7093	0,70885	0,7062	0,70685	0,707	0,7124	0,7114	0,71265	0,73325	0,72965	0,72455	0,7202	0,7197	0,7162	0,71505	0,71335	0,7126	0,713	



● TPR

TPR	PCTBagging																						
	0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %	0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %	
Porcentajes de Consoli	Balanceadas y 99% coverage										100 muestras Bootstrap												
Datasets	1	2	3	4	5	6	7	8	9	10	11												
abalone19 (50)	0.5476	0.56	0.5338	0.5371	0.531	0.5143	0.5114	0.52	0.5076	0.501	0.4948	0	0	0	0	0	0	0	0	0	0	0	0
abalone9-18 (50)	0.6997	0.71	0.7033	0.7036	0.7056	0.7083	0.7058	0.7033	0.7056	0.7011	0.7011	0.2697	0.3267	0.3342	0.3414	0.3481	0.3339	0.3339	0.3389	0.3339	0.3292	0.3267	0.3267
ecoli-0 (50)	0.9972	0.9972	0.9972	0.9972	0.9972	0.9902	0.9902	0.9902	0.9902	0.9902	0.9902	1	1	1	1	1	0.9916	0.9916	0.9916	0.9916	0.9916	0.9916	0.9916
ecoli1 (50)	0.9508	0.9496	0.9508	0.9418	0.9367	0.9278	0.9253	0.9201	0.9176	0.9138	0.9125	0.7665	0.7666	0.7506	0.7637	0.7782	0.7753	0.7857	0.7842	0.7802	0.7842	0.7802	0.7802
ecoli2 (50)	0.8615	0.8615	0.8536	0.8536	0.8536	0.8404	0.8404	0.8364	0.8344	0.8156	0.8156	0.7616	0.7693	0.7638	0.7502	0.7464	0.7291	0.7231	0.7211	0.7307	0.7209	0.7151	0.7151
ecoli3 (50)	0.9057	0.9057	0.9114	0.92	0.9171	0.9057	0.9057	0.8971	0.8829	0.88	0.88	0.5429	0.5743	0.5686	0.5914	0.5857	0.5829	0.5657	0.5629	0.56	0.5629	0.56	0.56
ecoli4 (50)	0.88	0.88	0.86	0.86	0.86	0.845	0.845	0.845	0.845	0.845	0.845	0.72	0.715	0.74	0.74	0.73	0.695	0.695	0.695	0.685	0.665	0.665	0.665
glass-0-1-2-3 (50)	0.9216	0.9236	0.9216	0.9176	0.9042	0.9002	0.9002	0.892	0.8902	0.8824	0.8824	0.8707	0.8729	0.8691	0.8673	0.8595	0.832	0.8376	0.8338	0.832	0.8342	0.8285	0.8285
glass0 (50)	0.8786	0.8886	0.88	0.87	0.8614	0.8471	0.8471	0.8386	0.8343	0.8314	0.8229	0.7886	0.8	0.7857	0.7657	0.7514	0.7271	0.7243	0.7157	0.7129	0.72	0.7286	0.7286
glass1 (50)	0.6994	0.6783	0.6738	0.6752	0.6673	0.6673	0.6612	0.6625	0.6624	0.6597	0.657	0.6074	0.6181	0.6202	0.6349	0.6205	0.6167	0.6261	0.6275	0.6222	0.6248	0.6208	0.6208
glass2 (50)	0.8417	0.835	0.8183	0.8067	0.8083	0.7917	0.7917	0.7683	0.7733	0.735	0.735	0.6383	0.1033	0.1117	0.1115	0.1567	0.1683	0.1967	0.1967	0.1967	0.1967	0.1867	0.1867
glass4 (50)	0.8967	0.8967	0.8967	0.88	0.88	0.88	0.88	0.88	0.85	0.85	0.85	0.5267	0.5433	0.5333	0.53	0.54	0.5533	0.5533	0.5433	0.53	0.53	0.5233	0.5233
glass5 (50)	1	1	1	1	1	1	1	1	1	1	1	1	0.75	0.75	0.8	0.83	0.94	0.94	0.92	0.92	0.85	0.85	0.85
glass6 (50)	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.82	0.82	0.8553	0.8587	0.8587	0.834	0.834	0.834	0.802	0.7987	0.7987
haberman (50)	0.5673	0.5393	0.5588	0.5773	0.5724	0.57	0.5738	0.569	0.5701	0.5713	0.5713	0.2553	0.2922	0.2908	0.2626	0.2638	0.2638	0.2614	0.2578	0.2578	0.2578	0.259	0.259
iris0 (50)	1	1	1	1	1	0.968	0.968	0.968	0.968	0.968	0.968	1	1	1	1	1	0.968	0.968	0.968	0.968	0.968	0.968	0.968
new-thyroid1 (50)	0.9629	0.9629	0.96	0.9429	0.9429	0.94	0.94	0.94	0.9314	0.9314	0.9314	0.8857	0.8771	0.86	0.8486	0.8514	0.8543	0.8543	0.8514	0.8457	0.84	0.8429	0.8429
new-thyroid2 (50)	0.96	0.96	0.96	0.9543	0.9543	0.9457	0.9457	0.9457	0.9343	0.9343	0.9343	0.8943	0.8886	0.8686	0.8714	0.8743	0.8714	0.8714	0.8714	0.8714	0.8686	0.8714	0.8714
page-blocks (50)	0.9667	0.9644	0.9619	0.9612	0.9581	0.9576	0.9564	0.9553	0.9535	0.9505	0.9496	0.8655	0.8623	0.8637	0.8599	0.8567	0.8523	0.8507	0.8494	0.8471	0.8447	0.8447	0.8447
svm (50)	0.7742	0.7563	0.76	0.7492	0.7279	0.7253	0.7204	0.716	0.7148	0.7115	0.7096	0.6049	0.6265	0.5981	0.5891	0.591	0.5895	0.5835	0.5784	0.578	0.5824	0.5817	0.5817
segment0 (50)	0.9845	0.9839	0.9845	0.9845	0.9845	0.9845	0.9845	0.9845	0.9842	0.9842	0.9833	0.9833	0.969	0.9687	0.9687	0.9684	0.9684	0.9684	0.9684	0.9659	0.9656	0.9681	0.9681
svm (50)	0.9663	0.9663	0.9663	0.9653	0.9663	0.9593	0.9563	0.9523	0.9473	0.9468	0.9463	0.9255	0.9215	0.9186	0.91	0.8969	0.8844	0.8768	0.8793	0.8754	0.8749	0.8739	0.8739
svm (50)	0.8336	0.8335	0.8243	0.8183	0.8101	0.7944	0.7769	0.7722	0.7667	0.758	0.7493	0.4772	0.4981	0.5027	0.5051	0.5017	0.51	0.5129	0.5027	0.4944	0.4944	0.4976	0.4976
svm (50)	0.9761	0.9747	0.9729	0.9674	0.9615	0.9596	0.9592	0.9574	0.9551	0.95	0.9482	0.9523	0.9413	0.9376	0.9225	0.9188	0.917	0.9179	0.9179	0.9179	0.9026	0.9025	0.9025
svm (50)	0.8552	0.8343	0.8306	0.8155	0.8169	0.8032	0.7962	0.7906	0.7878	0.7821	0.7764	0.4306	0.4622	0.4828	0.5053	0.5144	0.5224	0.5214	0.5147	0.511	0.5082	0.5073	0.5073
svm (50)	0.9778	0.9778	0.9756	0.9744	0.9744	0.9733	0.9733	0.9711	0.9711	0.9711	0.9711	0.9378	0.9256	0.9267	0.9211	0.9133	0.9067	0.9033	0.9022	0.9089	0.9122	0.92	0.92
svm (50)	0.9636	0.9657	0.9644	0.959	0.9594	0.9577	0.9569	0.9561	0.9556	0.9552	0.9556	0.9557	0.9506	0.9531	0.9498	0.9406	0.936	0.9372	0.938	0.9359	0.9347	0.9351	0.9351
svm (50)	0.615	0.615	0.615	0.63	0.625	0.635	0.63	0.63	0.635	0.63	0.63	0.02	0.02	0.02	0.05	0.05	0.215	0.215	0.22	0.245	0.245	0.245	0.245
svm (50)	0.7068	0.6944	0.7002	0.7017	0.7042	0.7005	0.698	0.6931	0.6896	0.6875	0.6844	0.4877	0.4917	0.4823	0.4872	0.4868	0.484	0.4798	0.4807	0.4803	0.4807	0.4812	0.4812
svm (50)	0.9289	0.9289	0.9289	0.9258	0.9246	0.9228	0.9228	0.9221	0.9203	0.9173	0.9173	0.7844	0.7846	0.7846	0.7607	0.7497	0.746	0.7393	0.7368	0.7393	0.7418	0.7449	0.7449
svm (50)	0.8893	0.8893	0.8836	0.8855	0.8855	0.8856	0.8836	0.8704	0.8665	0.8567	0.8547	0.3004	0.3022	0.316	0.3142	0.3195	0.3078	0.3002	0.2994	0.306	0.306	0.308	0.308
svm (50)	0.9775	0.9775	0.9775	0.98	0.98	0.9731	0.9731	0.9731	0.9681	0.9681	0.9681	0.7744	0.7744	0.7722	0.7694	0.7492	0.742	0.7403	0.7225	0.7247	0.7247	0.7239	0.7239
svm (50)	0.8543	0.8543	0.8543	0.8543	0.8543	0.8343	0.8343	0.8343	0.8114	0.8114	0.8114	0.4714	0.4657	0.4714	0.4857	0.4829	0.4914	0.4829	0.4629	0.4486	0.44	0.4371	0.4371
Average	0.8708	0.8685	0.866	0.8638	0.8613	0.8547	0.853	0.85	0.8461	0.842	0.8404	0.6493	0.6579	0.6587	0.6587	0.6588	0.6585	0.661	0.6606	0.6571	0.6548	0.652	0.6517

PCT

50% Atributos	100 muestras modo RM																					
	0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %	0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %
Balanceadas y 99% coverage	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
0.5986	0.611	0.5705	0.551	0.539	0.509	0.5	0.489	0.481	0.4576	0.4448	0	0	0	0	0	0	0	0	0	0	0	0
0.6761	0.645	0.6425	0.6575	0.6531	0.6533	0.6533	0.6458	0.6433	0.6431	0.6406	0.1264	0.2003	0.2097	0.2275	0.2375	0.2539	0.2656	0.2703	0.2728	0.2775	0.28	0.28
0.9993	0.9986	0.9986	0.9986	0.9986	0.9916	0.9916	0.9881	0.9881	0.9874	0.9874	1	1	1	1	1	1	0.9916	0.9916	0.9916	0.9916	0.9916	0.9916
0.9395	0.9383	0.9396	0.9304	0.9278	0.9239	0.9226	0.9251	0.9277	0.9238	0.9225	0.7551	0.7953	0.7795	0.7662	0.7572	0.7718	0.7757	0.7821	0.7427	0.744	0.744	0.744
0.8671	0.8671	0.8747	0.8765	0.8765	0.8518	0.8518	0.8438	0.8213	0.8213	0.8213	0.7196	0.7389	0.7525	0.7395	0.7413	0.7276	0.7276	0.7433	0.7491	0.7475	0.7475	0.7475
0.94	0.94	0.9371	0.9486	0.94	0.9257	0.9257	0.9229	0.8914	0.8714	0.8714	0.3914	0.4286	0.4743	0.5114	0.5429	0.5514	0.5514	0.5457	0.5343	0.5343	0.5343	0.5343
0.9	0.9	0.875	0.875	0.875	0.84	0.84	0.84	0.84	0.84	0.84	0.52	0.52	0.61	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.63	0.63
0.9565	0.9485	0.9489	0.9449	0.9369	0.9216	0.9196	0.9136	0.9116	0.906	0.906	0.8473	0.8691	0.8749	0.8729	0.8596	0.834	0.8282	0.8262	0.8242	0.8242	0.8242	0.8278
0.87	0.8871	0.88	0.8671	0.8471	0.83	0.8257	0.8157	0.81	0.8086	0.8129	0.7643	0.7886	0.7986	0.7929	0.77	0.7629	0.76	0.7614	0.7586	0.7657	0.7686	0.7686
0.6994	0.6602	0.6518	0.6769	0.71	0.7179	0.714	0.7103	0.7102	0.6972	0.6877	0.566	0.5483	0.5536	0.5738	0.5805	0.5778	0.5712	0.5726	0.5727	0.5687	0.5687	0.5687
0.8667	0.8533	0.8417	0.8333	0.8217	0.79	0.79	0.7633	0.7633	0.7183	0.7183	0	0.005	0.0417	0.0833	0.155	0.2317						

**RSM**

75% Atributos																								
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %			
Balanceadas y 99% coverage											100 muestras modo RSM													
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55			
0,5643	0,5671	0,5414	0,5386	0,5252	0,5186	0,5181	0,5019	0,5014	0,4914	0,4848	0	0	0	0	0	0	0	0	0	0	0			
0,6831	0,6664	0,6619	0,6675	0,6675	0,6722	0,6722	0,67	0,67	0,6722	0,6697	0,2503	0,2647	0,2603	0,2603	0,2653	0,2656	0,2656	0,2656	0,2681	0,2728	0,2753			
0,9979	0,9979	0,9979	0,9979	0,9979	0,9895	0,9895	0,9895	0,9895	0,9888	0,9888	1	1	1	1	1	0,9916	0,9916	0,9916	0,9916	0,9916	0,9916			
0,9495	0,9469	0,9442	0,9338	0,9241	0,9163	0,9176	0,9086	0,9098	0,9098	0,9072	0,8058	0,7952	0,7884	0,7419	0,7368	0,747	0,751	0,7548	0,7413	0,744	0,744			
0,8613	0,8595	0,85	0,8558	0,852	0,85	0,85	0,8402	0,8304	0,8175	0,8175	0,7413	0,7447	0,7447	0,7373	0,7411	0,7333	0,7333	0,7453	0,7511	0,7475	0,7475			
0,9371	0,9371	0,9429	0,9486	0,9229	0,9171	0,9171	0,9143	0,8771	0,8543	0,8543	0,4486	0,4686	0,4971	0,5229	0,5286	0,5314	0,5286	0,5257	0,5229	0,5229	0,5229			
0,885	0,885	0,85	0,85	0,85	0,84	0,84	0,84	0,84	0,835	0,835	0,62	0,62	0,64	0,65	0,65	0,64	0,64	0,64	0,64	0,63	0,63			
0,9427	0,9427	0,9507	0,9389	0,9273	0,9022	0,9022	0,9004	0,9024	0,908	0,908	0,8515	0,8476	0,846	0,844	0,838	0,8258	0,8242	0,8242	0,8242	0,8242	0,8278			
0,8729	0,8857	0,8843	0,87	0,8543	0,84	0,8314	0,8286	0,8171	0,8143	0,8057	0,78	0,7814	0,7843	0,7714	0,7714	0,7643	0,7629	0,7671	0,7686	0,7686	0,7686			
0,7074	0,6704	0,6789	0,6737	0,6844	0,6887	0,6848	0,6729	0,6742	0,6715	0,6675	0,5593	0,5635	0,5715	0,5817	0,5844	0,5804	0,5766	0,5753	0,5767	0,5793	0,578			
0,8467	0,8333	0,8083	0,8017	0,7883	0,7617	0,7617	0,735	0,735	0,665	0,665	0,0533	0,11	0,1467	0,1983	0,2033	0,2133	0,2133	0,2133	0,2133	0,2133	0,2067			
0,9133	0,9133	0,9133	0,91	0,91	0,91	0,91	0,91	0,91	0,8567	0,8567	0,56	0,5733	0,5567	0,5733	0,5467	0,5467	0,5467	0,5467	0,5467	0,5467	0,5467			
1	1	1	1	1	1	1	1	1	1	1	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85			
0,8967	0,8967	0,8967	0,8967	0,8967	0,8967	0,8967	0,8967	0,8967	0,8967	0,8967	0,7847	0,7913	0,8087	0,8227	0,8227	0,8227	0,8227	0,8227	0,8127	0,8127	0,8127			
0,5762	0,5309	0,5501	0,5795	0,5795	0,5799	0,5788	0,5737	0,5763	0,5763	0,5738	0,0332	0,0357	0,0993	0,2101	0,2101	0,2776	0,2776	0,2776	0,2776	0,2789	0,2789			
1	1	1	1	1	0,97	0,97	0,97	0,97	0,97	0,97	1	1	1	1	1	0,968	0,968	0,968	0,968	0,968	0,968			
0,9743	0,9743	0,9714	0,9486	0,9486	0,9486	0,9486	0,9486	0,9429	0,9429	0,9429	0,8686	0,8857	0,8829	0,88	0,8686	0,8686	0,8686	0,8657	0,8657	0,8657	0,8657			
0,9686	0,9686	0,9686	0,9457	0,9457	0,9343	0,9343	0,9343	0,9286	0,9286	0,9286	0,8771	0,8857	0,8771	0,8771	0,8771	0,88	0,88	0,8743	0,8714	0,8714	0,8714			
0,9694	0,9651	0,9637	0,9633	0,9592	0,9567	0,9565	0,9551	0,9535	0,9513	0,9499	0,858	0,8533	0,8542	0,8556	0,8548	0,8515	0,8514	0,8512	0,8506	0,8505	0,8505			
0,7683	0,7694	0,7526	0,7389	0,7303	0,7217	0,712	0,7056	0,7012	0,6982	0,6941	0,5235	0,5736	0,5583	0,5831	0,5834	0,5898	0,5917	0,5909	0,5898	0,5887	0,5895			
0,9854	0,9845	0,9824	0,9818	0,9805	0,9811	0,9818	0,9818	0,9818	0,9808	0,9808	0,969	0,9678	0,9666	0,9669	0,9663	0,9663	0,9663	0,9663	0,9663	0,9663	0,9663			
0,9708	0,9713	0,9689	0,9694	0,9663	0,9603	0,9588	0,9558	0,9518	0,9513	0,9483	0,9155	0,907	0,905	0,899	0,8954	0,8984	0,8869	0,8874	0,8884	0,8863	0,8889			
0,8354	0,8331	0,823	0,8179	0,8051	0,8042	0,7958	0,7935	0,7866	0,782	0,7759	0,434	0,4664	0,4792	0,4847	0,4741	0,4737	0,4714	0,4677	0,4682	0,4705	0,4715			
0,9803	0,9738	0,972	0,9674	0,9642	0,9629	0,961	0,9592	0,9605	0,9583	0,9546	0,9382	0,933	0,9316	0,9266	0,9206	0,9183	0,9178	0,9183	0,9174	0,9174	0,9174			
0,858	0,8476	0,842	0,8353	0,8268	0,8066	0,8014	0,7896	0,7849	0,782	0,775	0,4213	0,4501	0,4619	0,4822	0,4926	0,4921	0,4935	0,494	0,4941	0,4955	0,4955			
0,9778	0,9778	0,9733	0,9722	0,9722	0,9722	0,9722	0,97	0,97	0,97	0,97	0,9422	0,9367	0,9367	0,9367	0,9333	0,9333	0,9344	0,9378	0,9378	0,9378	0,9378			
0,9687	0,9674	0,9665	0,9627	0,9619	0,9615	0,9623	0,9602	0,959	0,959	0,959	0,9477	0,9423	0,9418	0,9406	0,9364	0,9335	0,9327	0,9327	0,9322	0,9318	0,9314			
0,62	0,62	0,615	0,615	0,62	0,615	0,61	0,6	0,61	0,615	0,615	0	0	0	0,005	0,005	0,02	0,02	0,02	0,02	0,02	0,02			
0,7124	0,6991	0,7028	0,7044	0,7075	0,6947	0,6951	0,6921	0,687	0,6809	0,6793	0,3874	0,4576	0,4736	0,4645	0,4636	0,4659	0,4659	0,4661	0,465	0,465	0,4636			
0,9258	0,9264	0,9276	0,9215	0,9203	0,9203	0,9191	0,9172	0,9136	0,9117	0,9117	0,8686	0,7961	0,8043	0,7955	0,7791	0,7804	0,7804	0,7754	0,7748	0,7748	0,7779			
0,8582	0,8405	0,8095	0,8485	0,876	0,8796	0,8736	0,8642	0,8564	0,8347	0,8327	0,026	0,1396	0,1825	0,2411	0,2609	0,2685	0,2665	0,2664	0,2664	0,2664	0,2644			
0,9775	0,9775	0,9797	0,9822	0,98	0,9756	0,9756	0,9756	0,9611	0,9611	0,9611	0,5594	0,7117	0,7656	0,7836	0,7858	0,7928	0,7861	0,7906	0,7906	0,7906	0,7906			
0,86	0,86	0,86	0,8629	0,8629	0,8343	0,8343	0,8343	0,8114	0,8114	0,8114	0,2057	0,3	0,3286	0,4	0,4029	0,4371	0,4286	0,44	0,44	0,4457	0,4457			
0,8741	0,8694	0,8651	0,8636	0,8608	0,8540	0,8525	0,8481	0,8426	0,8378	0,8361	0,6030	0,6258	0,6347	0,6450	0,6440	0,6460	0,6452	0,6457	0,6452	0,6454	0,6454			
<b>0,8741</b>	<b>0,8694</b>	<b>0,8651</b>	<b>0,8636</b>	<b>0,8608</b>	<b>0,854</b>	<b>0,8525</b>	<b>0,8481</b>	<b>0,8426</b>	<b>0,8378</b>	<b>0,8361</b>	<b>0,603</b>	<b>0,6258</b>	<b>0,6347</b>	<b>0,645</b>	<b>0,644</b>	<b>0,646</b>	<b>0,6452</b>	<b>0,6457</b>	<b>0,6452</b>	<b>0,6454</b>	<b>0,6453</b>			

**PC1**

50% Atributos																								
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %			
Balanceadas y 99% coverage											100 muestras modo RSM													
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77			
0,5705	0,5829	0,5686	0,5743	0,5519	0,5324	0,5395	0,5348	0,5381	0,5214	0,5062	0	0	0	0	0	0	0	0	0	0	0			
0,6786	0,6692	0,6739	0,6833	0,6881	0,6747	0,6725	0,6697	0,6675	0,6628	0,6606	0,2339	0,2347	0,2464	0,2556	0,2656	0,2722	0,2769	0,2842	0,2842	0,2889	0,2914			
0,9993	0,9993	0,9993	0,9993	0,9993	0,9909	0,9909	0,9909	0,9909	0,9909	0,9909	1	1	1	1	1	0,9916	0,9916	0,9916	0,9916	0,9916	0,9916			
0,9599	0,9613	0,9613	0,951	0,9509	0,9304	0,9291	0,9211	0,9212	0,9211	0,9185	0,7941	0,7914	0,7731	0,7547	0,7468	0,7558	0,7585	0,7597	0,7413	0,744	0,744			
0,8673	0,8673	0,8707	0,8649	0,8629	0,8498	0,8498	0,8436	0,836	0,8191	0,8191	0,7544	0,7582	0,7565	0,7549	0,7549	0,7355	0,7355	0,7453	0,7511	0,7475	0,7475			
0,9314	0,9314	0,9371	0,9457	0,9314	0,92	0,92	0,9171	0,8886	0,88	0,88	0,4457	0,4657	0,4829	0,4914	0,5314	0,5343	0,5371	0,5343	0,5257	0,52	0,52			
0,875	0,875	0,86	0,855	0,855	0,85	0,85	0,84	0,84	0,84	0,84	0,645	0,645	0,65	0,665	0,67	0,66	0,66	0,66	0,65	0,63	0,63			
0,9413	0,9413	0,9393	0,9335	0,9256	0,9078	0,9058	0,9016	0,8956	0,888	0,888	0,8555	0,8615	0,8615	0,8555	0,8438	0,836	0,8224	0,8242	0,8242	0,8242	0,8278			
0,8671	0,86	0,8586	0,8557	0,84	0,8186	0,8029	0,8	0,7986	0,7871	0,7943	0,78	0,78	0,7871	0,7814	0,7729	0,7643	0,7586	0,7557	0,7543	0,7614	0,7614			
0,7087	0,6863	0,6608	0,677	0,699	0,6852	0,6927	0,681	0,6782	0,6613	0,6583	0,5572	0,5538	0,5549	0,5844	0,5938	0,5857	0,578	0,5702	0,5753	0,5766	0,5766			
0,83	0,825	0,8783	0,8717	0,86	0,84	0,84	0,8133	0,7967	0,6883	0,6883	0,0067	0,055	0,0683	0,125	0,1933	0,22	0,2417	0,25	0,2433	0,2433	0,23			
0,9133	0,9133	0,9133	0,9267	0,9267	0,9267	0,9267	0,9267	0,86	0,86	0,86	0,56	0,58	0,5933	0,5867	0,5933	0,61	0,6033	0,5933	0,6	0,5533	0,5533			
1	1	1	1	1	1	1	1	1	1	1	0,74	0,74	0,83	0,83	0,83	0,93	0,93	0,93	0,93	0,85	0,85			
0,8967	0,8967	0,8967	0,89																					

RF

75% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balanceadas y 99% coverage											100 muestras modo RSM										
78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0.5576	0.5638	0.5548	0.539	0.5386	0.5314	0.5257	0.5348	0.5281	0.509	0.5024	0	0	0	0	0	0	0	0	0	0	0
0.6886	0.6722	0.6678	0.6897	0.6964	0.6819	0.6797	0.6722	0.6722	0.6653	0.6631	0.2575	0.2675	0.2725	0.265	0.2653	0.2681	0.2656	0.2656	0.2681	0.2753	0.2753
0.9986	0.9986	0.9986	0.9986	0.9986	0.9902	0.9902	0.9902	0.9902	0.9902	0.9902	1	1	1	1	1	0.9916	0.9916	0.9916	0.9916	0.9916	0.9916
0.9573	0.9587	0.9586	0.9483	0.9419	0.9315	0.9264	0.9264	0.9214	0.9188	0.9148	0.7682	0.7592	0.7512	0.7327	0.7187	0.7327	0.738	0.748	0.744	0.744	0.744
0.8713	0.8635	0.8613	0.8573	0.8573	0.844	0.844	0.8422	0.8402	0.8269	0.8249	0.7529	0.7567	0.7547	0.7509	0.7511	0.7471	0.7453	0.7473	0.7511	0.7475	0.7475
0.9343	0.9343	0.9343	0.9429	0.9371	0.92	0.92	0.9143	0.8829	0.8743	0.8743	0.4914	0.4971	0.5229	0.5286	0.5429	0.54	0.54	0.5343	0.5314	0.5314	0.5314
0.87	0.87	0.86	0.86	0.86	0.855	0.855	0.855	0.855	0.855	0.855	0.65	0.65	0.64	0.645	0.65	0.64	0.64	0.64	0.64	0.63	0.63
0.9331	0.9331	0.9353	0.9295	0.9235	0.906	0.904	0.902	0.898	0.896	0.896	0.8515	0.8456	0.8458	0.8498	0.8438	0.8258	0.8278	0.8278	0.8278	0.8278	0.8278
0.8657	0.8729	0.87	0.87	0.8471	0.8329	0.8314	0.8314	0.8229	0.8229	0.8186	0.78	0.7757	0.7743	0.7686	0.7586	0.7586	0.76	0.7643	0.7643	0.7643	0.7643
0.6912	0.6796	0.6808	0.7041	0.6926	0.6772	0.6784	0.6653	0.6733	0.6652	0.6693	0.5543	0.5738	0.5766	0.5778	0.5752	0.5765	0.5727	0.5727	0.5753	0.5767	0.5753
0.855	0.855	0.8333	0.8333	0.8333	0.8033	0.8033	0.77	0.7533	0.7283	0.7233	0.1517	0.1883	0.1867	0.2117	0.2167	0.2183	0.2183	0.2133	0.2133	0.2133	0.2067
0.9133	0.9133	0.9133	0.8933	0.8933	0.8933	0.8933	0.8933	0.8667	0.8667	0.8667	0.57	0.5833	0.59	0.59	0.57	0.57	0.57	0.5467	0.5467	0.5467	0.5467
1	1	1	1	1	1	1	1	1	1	1	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.8967	0.78	0.7907	0.808	0.8187	0.8227	0.8227	0.8227	0.8227	0.8127	0.8127	0.8127
0.5573	0.5131	0.5404	0.575	0.571	0.5578	0.5579	0.5543	0.5429	0.5504	0.5504	0.0779	0.0779	0.129	0.1771	0.1771	0.2559	0.2559	0.2559	0.2789	0.2789	0.2789
1	1	1	1	1	0.968	0.968	0.968	0.968	0.968	0.968	1	1	1	1	1	0.968	0.968	0.968	0.968	0.968	0.968
0.9571	0.9571	0.9543	0.9486	0.9486	0.9371	0.9371	0.9371	0.9343	0.9343	0.9343	0.86	0.8743	0.8714	0.8686	0.8657	0.8657	0.8657	0.8629	0.8629	0.8629	0.8629
0.9543	0.9543	0.9543	0.9343	0.9343	0.9257	0.9257	0.9257	0.9171	0.9086	0.9086	0.88	0.8886	0.8771	0.8771	0.8771	0.88	0.88	0.8743	0.8714	0.8714	0.8714
0.9678	0.9646	0.9605	0.9628	0.9589	0.9553	0.9558	0.9546	0.9522	0.9521	0.9497	0.8608	0.8551	0.8564	0.8544	0.8535	0.8515	0.8508	0.8512	0.8506	0.8501	0.8501
0.7693	0.7772	0.7552	0.7552	0.7291	0.728	0.7231	0.7183	0.7194	0.7161	0.712	0.5791	0.578	0.5746	0.5887	0.5805	0.5861	0.5906	0.5917	0.5891	0.5891	0.5887
0.9857	0.986	0.9848	0.9839	0.9827	0.983	0.983	0.983	0.9827	0.9824	0.9824	0.9693	0.9678	0.9669	0.9669	0.9663	0.9663	0.9663	0.9663	0.9663	0.9663	0.9663
0.9698	0.9724	0.9678	0.9648	0.9648	0.9598	0.9568	0.9528	0.9523	0.9528	0.9473	0.915	0.9035	0.9035	0.8944	0.8919	0.8889	0.8874	0.8879	0.8879	0.8879	0.8889
0.8244	0.8332	0.8185	0.8096	0.8013	0.7976	0.7912	0.7834	0.7737	0.7728	0.7641	0.4451	0.4616	0.4747	0.4756	0.4718	0.4705	0.4696	0.4669	0.4687	0.4706	0.4696
0.9748	0.9734	0.9748	0.9702	0.9675	0.9656	0.961	0.9597	0.9569	0.9533	0.9496	0.93	0.9321	0.928	0.9257	0.9211	0.9178	0.9178	0.9183	0.9174	0.9174	0.9174
0.858	0.8386	0.8368	0.8131	0.8112	0.797	0.7968	0.7797	0.7787	0.7741	0.7655	0.4403	0.462	0.4728	0.4874	0.4922	0.4908	0.4903	0.4945	0.4922	0.4922	0.4927
0.9778	0.9778	0.9767	0.9756	0.9756	0.9744	0.9744	0.9722	0.9722	0.9722	0.9722	0.9389	0.9389	0.9389	0.9367	0.9333	0.9333	0.9333	0.9344	0.9378	0.9378	0.9378
0.9657	0.9653	0.9661	0.9607	0.9602	0.9611	0.9615	0.9607	0.9594	0.9607	0.9607	0.9473	0.9414	0.9385	0.9364	0.9385	0.9339	0.9339	0.9335	0.9335	0.9327	0.9318
0.615	0.615	0.61	0.61	0.625	0.62	0.615	0.61	0.605	0.6	0.6	0	0	0	0.005	0.005	0.02	0.02	0.02	0.02	0.02	0.02
0.7024	0.6935	0.6923	0.6874	0.6853	0.6837	0.6854	0.6821	0.673	0.6728	0.6695	0.4594	0.4655	0.4722	0.4659	0.4689	0.4685	0.4671	0.4678	0.4689	0.468	0.4675
0.9258	0.9258	0.927	0.9221	0.9196	0.9172	0.9172	0.916	0.9154	0.9166	0.9166	0.7988	0.8042	0.803	0.7999	0.7828	0.7804	0.7822	0.7747	0.7748	0.7754	0.7797
0.866	0.8582	0.8502	0.8636	0.8776	0.8856	0.8836	0.8662	0.8642	0.8545	0.8545	0.1835	0.2291	0.2467	0.2645	0.2624	0.2724	0.2704	0.2684	0.2644	0.2644	0.2644
0.9775	0.9775	0.9775	0.9797	0.9797	0.9775	0.9775	0.9775	0.9658	0.9658	0.9658	0.7831	0.7872	0.7969	0.7969	0.7994	0.7906	0.7858	0.7883	0.7883	0.7906	0.7906
0.86	0.86	0.86	0.8629	0.8629	0.8371	0.8371	0.8371	0.8171	0.8171	0.8171	0.4371	0.44	0.4486	0.4457	0.4457	0.4486	0.4486	0.4429	0.4429	0.4457	0.4457
0.8710	0.8683	0.8658	0.8649	0.8628	0.8544	0.8529	0.8495	0.8440	0.8406	0.8389	0.6353	0.6408	0.6446	0.6471	0.6454	0.6464	0.6462	0.6452	0.6455	0.6455	0.6453
0.8709	0.8683	0.8658	0.8649	0.8628	0.8544	0.8529	0.8495	0.844	0.8406	0.8389	0.6353	0.6408	0.6446	0.6471	0.6454	0.6464	0.6462	0.6452	0.6455	0.6455	0.6453

## TNR

TNR	Porcentajes de Consolidación	PCTesting																							
		0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %		
		Balanceadas y 99% coverage										100 muestras Bootstrap													
Datasets		1	2	3	4	5	6	7	8	9	10	11													
abalone19 (50)	0.8212	0.8162	0.8138	0.8104	0.8105	0.8108	0.8104	0.8127	0.8173	0.8226	0.8281	1	1	1	1	1	1	1	1	1	1	1	1	1	1
abalone9-18 (50)	0.7959	0.7906	0.7849	0.7739	0.7749	0.7674	0.7678	0.7694	0.7688	0.7689	0.7702	0.997	0.9916	0.9891	0.9881	0.9859	0.9838	0.9827	0.9819	0.9813	0.9811	0.9813	0.9813	0.9813	0.9813
ecoli_0 (50)	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9638	0.9638	0.9638	0.9638	0.9638	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742	0.9742
ecoli1 (50)	0.8495	0.8464	0.8452	0.8533	0.8548	0.8606	0.8653	0.8714	0.8726	0.8749	0.8765	0.9456	0.9417	0.9448	0.9355	0.9328	0.9336	0.9328	0.9316	0.9324	0.9324	0.9343	0.9343	0.9343	0.9343
ecoli2 (50)	0.9182	0.9186	0.9137	0.9147	0.9144	0.92	0.92	0.9218	0.9221	0.9423	0.9423	0.9817	0.9771	0.9746	0.975	0.9739	0.9743	0.9746	0.9732	0.9725	0.9722	0.9722	0.9722	0.9722	0.9722
ecoli3 (50)	0.8455	0.8455	0.8445	0.8438	0.8458	0.8508	0.8508	0.8511	0.8651	0.8681	0.8681	0.9794	0.9778	0.9746	0.9711	0.9701	0.9675	0.9665	0.9638	0.9638	0.9635	0.9631	0.9631	0.9631	0.9631
ecoli4 (50)	0.9104	0.9104	0.8902	0.8886	0.8877	0.8823	0.8823	0.8832	0.8911	0.9184	0.9184	0.9886	0.9886	0.9858	0.9842	0.9835	0.9835	0.9835	0.9835	0.9839	0.9857	0.9857	0.9857	0.9857	0.9857
glass-0-1-2-3 (50)	0.8953	0.8947	0.8941	0.8947	0.8984	0.8971	0.8921	0.9026	0.902	0.9032	0.9032	0.9259	0.9534	0.9547	0.9572	0.959	0.9596	0.9584	0.9571	0.9565	0.9547	0.9541	0.9541	0.9541	0.9541
glass0 (50)	0.7559	0.7434	0.74	0.7449	0.754	0.7637	0.7671	0.7713	0.7727	0.7762	0.7832	0.8882	0.868	0.859	0.8519	0.8555	0.857	0.8549	0.857	0.8577	0.8598	0.8612	0.8612	0.8612	0.8612
glass1 (50)	0.7574	0.7602	0.7646	0.7545	0.7458	0.7357	0.7365	0.7452	0.7414	0.7479	0.7544	0.9044	0.8834	0.861	0.8364	0.8292	0.8305	0.8248	0.8169	0.8169	0.8147	0.8232	0.8232	0.8232	0.8232
glass2 (50)	0.6842	0.6832	0.671	0.6745	0.6785	0.6776	0.6792																		











75% Atributos

Table with 22 columns representing percentiles (0.00% to 100.00%) and 30 rows of data. The table is split into two main sections: 'Balancedas y 99% coverage' and '100 muestras mdo RSM'. The data is presented as a grid of numerical values.

• Kappa

KAPPA table with 21 columns (0.00% to 100.00%) and 36 rows of datasets. The table is split into two main sections: 'Balancedas y 99% coverage' and '100 muestras Bootstrap'. The datasets listed include abalone19, abalone9-18, ecol0-0, ecol1, ecol2, ecol3, ecol4, glass-0-1-2-3, glass0, glass1, glass2, glass4, glass5, glass6, hbsurmanlab, iso0, new-thyroid1, new-thyroid2, page-blocks0, pima1mb, segment0, ehicle0, ehicle1, ehicle2, ehicle3, owe0, wisconsinlab, yeast-2, yeast1, yeast3, yeast4, yeast5, and yeast6. The table concludes with an 'Average' row.









RSM

75% Atributos																					
0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%	0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%
Balancedas y 99% coverage											100 muestras modo RSM										
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
1	2,84	4,04	5,66	7,12	9,06	9,9	11,7	12,86	14,26	15,74	1	1	1	1	1	1	1	1	1	1	1
1	1,64	2,24	3	3,28	5,2	5,4	6	6,34	7,26	7,58	1	2,64	4,06	5,46	6,8	8,82	9,94	11,46	12,68	14,18	15,72
1	1,04	1,12	1,16	1,24	3,16	3,16	3,28	3,28	3,4	3,4	1	1	1	1	1	3	3	3	3	3	3
1	2	2,48	3,4	3,92	5,64	6,08	7	7,52	8,4	9	1	2,04	2,68	4,2	4,8	5,96	6,44	7,4	8,56	9,56	10,24
1	1,16	2,6	3,16	3,48	4,76	4,76	5,24	5,64	7,24	7,24	1	2,28	3,12	4,28	5,12	6,56	7,08	8,52	9,08	10,52	11,2
1	1	1,76	2,92	3,32	3,76	3,76	4,16	5,32	6,08	6,08	1	2,36	4,08	5,36	7	8,68	9,56	11,28	12,48	14,28	15,56
1	1	2,08	2,32	2,56	4,08	4,08	4,32	4,56	5,64	5,64	1	1	1,92	3	3,12	3,92	3,92	4,04	5,12	6,04	6,04
1	1,08	2,12	2,76	3,28	4,2	4,2	4,8	5,36	6,48	6,48	1	2,28	2,92	4,28	5,04	6,48	7,32	8,24	9,44	10,24	11,36
1	2,84	4,16	6,04	7,68	9,64	10,88	12,6	14,4	15,8	17,56	1	3	4,88	7,2	9,08	11,4	12,92	14,96	17,12	19,16	21
1	3,44	5,64	8,4	10,68	13,4	15,24	17,68	20,28	22,64	24,92	1	2,88	4,48	6,48	8,24	10,64	12	13,92	15,76	17,52	19,24
1	1,16	2,64	3,16	4,04	4,8	4,8	5,84	6,2	7,84	7,84	1	2,32	3,48	4,36	5,72	7,28	8,04	9,44	10,28	11,48	12,76
1	1	1,04	2,88	2,88	3,04	3,04	3,04	4,88	4,92	4,92	1	1,56	3	3,56	4,72	5,56	5,72	7,28	7,44	9,28	9,44
1	1	1	1,32	1,32	3	3	3	3,32	3,32	3,32	1	1	3	3	3	5	5	5	5	7	7
1	1	1,28	2,44	2,44	3,28	3,28	3,28	4,44	4,72	4,72	1	1,16	1,96	3,12	3,32	4,12	4,12	4,48	5,48	6,44	6,44
1	2,48	3,64	5,04	6,24	8,28	8,76	10,32	11,36	12,88	14	1	1,08	1,52	2,12	2,24	3,12	3,16	3,32	3,88	4,36	4,4
1	1	1	1	1	3	3	3	3	3	3	1	1	1	1	1	3	3	3	3	3	3
1	1	1,16	2,88	2,88	3,16	3,16	3,16	4,88	5,04	5,04	1	1,52	2,56	3,52	4,4	5,08	5,08	6,48	6,92	8,48	8,48
1	1	1,2	2,8	2,8	3,2	3,2	3,2	4,8	5	5	1	1,48	2,48	3,48	4,12	4,96	4,96	6,08	6,6	8,08	8,08
1	5,16	9,04	12,96	16,92	21,52	25	29,12	32,88	36,92	40,92	1	8,32	15,44	22,72	29,84	37,72	44,48	51,8	58,88	66,2	73,32
1	7,44	13,48	19,84	25,96	32,8	38,68	45	51,16	57,4	63,64	1	5,12	9,28	13,2	17,36	21,88	25,28	29,6	33,36	37,68	41,64
1	1,84	3,24	3,84	5,64	6,48	6,8	8,72	9,2	10,72	11,44	1	3	4,48	6,04	7,8	9,88	10,72	12,6	14,04	15,64	17,52
1	3	4,4	6,2	7,88	10,08	11,28	13,04	14,76	16,24	18,16	1	5,04	8,52	12,32	16,16	20,4	23,56	27,52	31,2	34,8	38,72
1	6,52	11,92	17,44	22,6	28,72	33,8	39,12	44,48	50,04	55,4	1	9,52	17,68	26,04	34,32	43,36	51,12	59,6	67,76	76,12	84,44
1	3,08	5,68	8,2	10,48	13,04	14,96	17,24	19,76	22,36	24,44	1	3,84	6,64	9,64	12,2	15,6	18	20,76	23,56	26,56	29,2
1	6,36	11,52	16,92	22,08	27,8	32,6	37,92	43,16	48,48	53,68	1	9,88	18,44	27,28	35,76	45,24	53,48	62,24	70,8	79,64	88,24
1	1,44	2,48	3,44	4,04	4,92	5,04	5,96	6,6	7,96	8,08	1	3	3,28	5,08	6,48	8,48	9,28	10,68	12,48	12,76	14,76
1	2,92	3,52	5,16	6,08	8	8,8	10,4	11,36	12,64	13,88	1	2,88	4,72	6,4	8,24	10,56	12	13,92	15,52	17,44	19,24
1	1,48	1,88	3,04	3,24	4,36	4,52	5,04	5,88	6,6	6,76	1	1,04	1,6	1,68	1,76	2,4	2,4	2,52	2,56	3,16	3,16
1	9	16,68	24,92	32,64	41,08	48,32	56,36	64,28	72,28	79,96	1	7	12,6	18,76	24,52	30,92	36,4	42,32	48,32	54,08	59,92
1	1,52	1,84	3,52	3,92	4,44	4,52	5,28	6,6	7,28	7,44	1	3,16	4,88	7,2	9,04	11,64	13,08	15,08	17,24	19,12	21,12
1	1,64	2,92	3,56	4,44	5,56	5,8	7,08	7,32	9	9,24	1	2,8	4,12	5,6	7,08	9,04	10	11,72	12,96	14,52	16,08
1	1	1,16	2,72	2,76	3,16	3,16	3,2	4,76	4,92	4,92	1	2,84	4,08	5,4	7,04	9	10,04	11,8	13	14,36	16,08
1	1,04	1,04	1,36	1,36	3,08	3,08	3,12	3,4	3,44	3,44	1	2,8	3,28	4,88	5,48	7,32	8,16	9,56	10,36	11,64	12,64
1,0000	2,4582	4,0000	5,8624	7,2788	9,3848	10,4867	12,0673	13,7588	15,4606	16,7539	1,0000	3,0861	5,0964	7,2321	9,1758	11,7582	13,3412	15,4733	17,4206	19,6164	21,5164
<b>1</b>	<b>2,4582</b>	<b>4</b>	<b>5,8624</b>	<b>7,2788</b>	<b>9,3848</b>	<b>10,4867</b>	<b>12,0673</b>	<b>13,7588</b>	<b>15,4606</b>	<b>16,7539</b>	<b>1</b>	<b>3,0861</b>	<b>5,0964</b>	<b>7,2321</b>	<b>9,1758</b>	<b>11,7582</b>	<b>13,3412</b>	<b>15,4733</b>	<b>17,4206</b>	<b>19,6164</b>	<b>21,5164</b>

PCT

50% Atributos																					
0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%	0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%
Balancedas y 99% coverage											100 muestras modo RSM										
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
1	2,84	4,18	6,02	7,5	9,62	10,54	12,34	13,7	15,22	16,7	1	1	1	1	1	1	1	1	1	1	1
1	1,44	1,74	2,36	2,52	4,42	4,54	4,9	5,22	5,74	5,98	1	2,68	4,1	5,5	6,88	8,9	9,92	11,58	12,74	14,18	15,72
1	1	1,12	1,12	1,12	3,12	3,12	3,12	3,12	3,24	3,24	1	1	1	1	1	3	3	3	3	3	3
1	2,04	2,52	3,64	4	5,68	6,04	6,96	7,52	8,56	9,04	1	2,04	2,68	4,2	4,8	5,96	6,44	7,4	8,56	9,56	10,24
1	1,16	2,56	3,16	3,52	4,72	4,76	5,24	5,72	7,24	7,28	1	2,28	3,12	4,28	5,12	6,56	7,08	8,52	9,08	10,52	11,2
1	1	1,8	2,88	3,2	3,8	3,8	4,12	5,2	6	6	1	2,36	4	5,32	6,92	8,48	9,2	10,88	12,12	13,84	15,12
1	1	2,08	2,36	2,48	4,08	4,08	4,2	4,48	5,56	5,56	1	1	1,92	3	3,12	3,92	3,92	4,04	5,12	6,04	6,04
1	1,16	2,08	2,84	3,28	4,24	4,32	4,84	5,52	6,52	6,6	1	2,32	2,92	4,32	5,12	6,56	7,36	8,36	9,56	10,36	11,48
1	2,96	4,12	6,04	7,48	9,68	10,92	12,56	14,28	15,64	17,4	1	3,04	4,96	7,24	9,12	11,48	13	15,08	17,16	19,28	21,12
1	3,24	5,68	8,08	10,4	13,08	14,88	17,32	19,6	22,16	24,28	1	2,96	4,72	6,72	8,52	10,96	12,52	14,48	16,32	18,24	20,04
1	1,2	2,6	3,2	4,12	4,8	4,8	5,92	6,32	7,92	7,92	1	2,4	3,48	4,48	5,92	7,52	8,16	9,64	10,6	11,72	13,08
1	1	1	2,8	2,8	3	3	3	4,8	4,8	4,8	1	1,48	3	3,48	4,68	5,48	5,56	7,16	7,24	9,16	9,24
1	1	1	1,32	1,32	3	3	3	3,32	3,32	3,32	1	1	3	3	3	5	5	5	5	7	7
1	1	1,4	2,2	2,2	3,4	3,4	3,4	4,2	4,6	4,6	1	1,2	2,04	3,16	3,32	4,24	4,24	4,6	5,52	6,56	6,56
1	2,4	3,56	5,04	6,04	8,08	8,84	10,08	11,32	12,72	13,88	1	1,08	1,52	2,12	2,24	3,12	3,16	3,32	3,88	4,36	4,4
1	1	1,12	1,12	1,12	3,12	3,12	3,12	3,12	3,24	3,24	1	1	1	1	1	3	3	3	3	3	3
1	1	1,4	2,88	2,88	3,4	3,4	3,4	4,88	5,28	5,28	1	1,32	2,48	3,32	4,28	4,8	4,8	6,08	6,6	8,08	8,08
1	1	1,16	2,88	2,88	3,16	3,16	3,16	4,88	5,04	5,04	1	1,36	2,4	3,36	4,08	4,76	4,76	5,84	6,44	7,84	7,84
1	5,28	9,44	13,68	17,72	22,48	26,2	30,36	34,48	38,76	42,92	1	8,28	15,44	22,8	30	37,88	44,52	51,96	59,08	66,48	73,52
1	6,96	12,72	18,52	24,28	30,6	36	41,88	47,56	53,44	59,28	1	4,88	8,88	12,64	16,6	20,88	24,08	28,16	31,8	35,92	39,68
1	1,88	3,32	3,96	5,8	6,68	7,08	8,96	9,56	11,04	11,88	1	3	4,48	6,04	7,8	9,88	10,72	12,6	14,04	15,64	17,52
1	3,04	4,4	6,4	8,04	10,44	11,72	13,48	15,36	16,84	18,76	1	5,04	8,52	12,32	16,2	20,44	23,52	27,52	31,2	34,8	38,72
1	6,48	11,6	17,2																		

TRF

		75% Atributos																					
		0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
		Balanceadas y 99% coverage										100 muestras modo RSM											
		78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
1	2,84	3,9	5,72	7,12	9,16	9,74	11,62	12,86	14,26	15,62	1	1	1	1	1	1	1	1	1	1	1	1	
1	1,64	1,9	2,74	2,96	4,8	4,92	5,6	5,76	6,5	6,66	1	2,64	4,06	5,46	6,76	8,78	9,9	11,42	12,58	14,12	15,62	15,62	
1	1	1	1	1	3	3	3	3	3	3	1	1	1	1	1	1	3	3	3	3	3	3	
1	2,04	2,64	3,72	4,36	5,92	6,2	7,36	7,92	9,04	9,56	1	2,04	2,68	4,2	4,8	5,96	6,44	7,4	8,56	9,56	10,24	10,24	
1	1,32	2,72	3,32	3,44	5,08	5,12	5,48	5,84	7,48	7,56	1	2,28	3,12	4,28	5,12	6,56	7,08	8,52	9,08	10,52	11,2	11,2	
1	1	1,68	2,92	3,16	3,68	3,68	3,92	5,16	5,84	5,84	1	2,36	4,12	5,4	7,04	8,68	9,56	11,28	12,48	14,32	15,6	15,6	
1	1	2,08	2,28	2,36	4,08	4,08	4,16	4,36	5,44	5,44	1	1	1,92	3	3,12	3,92	3,92	4,04	5,12	6,04	6,04	6,04	
1	1,12	2	2,88	3,2	4,12	4,16	4,56	5,36	6,32	6,36	1	2,28	2,92	4,28	5,04	6,48	7,32	8,24	9,44	10,24	11,36	11,36	
1	2,84	4,28	5,92	7,64	9,72	10,8	12,56	14,16	15,64	17,44	1	3	4,88	7,2	9,08	11,36	12,88	14,92	17,08	19,12	20,96	20,96	
1	3,28	5,64	8,08	10,32	13,08	15,08	17,36	19,76	22,16	24,4	1	2,88	4,44	6,48	8,2	10,6	11,96	13,84	15,72	17,44	19,16	19,16	
1	1,2	2,56	3,2	4,04	4,8	4,84	5,8	6,32	7,8	7,88	1	2,32	3,48	4,36	5,72	7,28	8,04	9,44	10,28	11,48	12,76	12,76	
1	1	1,04	2,84	2,84	3,04	3,04	3,04	4,84	4,88	4,88	1	1,56	3	3,56	4,72	5,56	5,68	7,28	7,4	9,28	9,4	9,4	
1	1	1	1,36	1,36	3	3	3	3,36	3,36	3,36	1	1	3	3	3	5	5	5	5	7	7	7	
1	1	1,44	2,48	2,48	3,44	3,44	3,44	4,48	4,92	4,92	1	1,16	1,96	3,12	3,28	4,12	4,12	4,44	5,44	6,4	6,4	6,4	
1	2,4	3,56	5,04	6,04	8,08	8,84	10,08	11,32	12,72	13,88	1	1,08	1,52	2,12	2,24	3,12	3,16	3,32	3,88	4,36	4,4	4,4	
1	1	1	1	1	3	3	3	3	3	3	1	1	1	1	1	3	3	3	3	3	3	3	
1	1	1,28	2,8	2,84	3,28	3,28	3,32	4,84	5,12	5,12	1	1,48	2,52	3,48	4,36	5	5	6,36	6,84	8,36	8,36	8,36	
1	1	1,12	2,8	2,8	3,12	3,12	3,12	4,8	4,92	4,92	1	1,48	2,48	3,48	4,12	4,96	4,96	6,08	6,6	8,08	8,08	8,08	
1	5,04	8,92	12,96	16,92	21,44	25,04	29,04	33,04	36,96	40,96	1	8,36	15,48	22,8	29,88	37,84	44,6	51,92	59	66,36	73,48	73,48	
1	7,44	13,28	19,72	25,8	32,56	38,24	44,64	50,76	56,92	63,04	1	5,12	9,28	13,2	17,36	21,88	25,28	29,6	33,36	37,68	41,64	41,64	
1	1,92	3,28	3,92	5,72	6,64	7	8,92	9,44	10,92	11,72	1	3	4,48	6,04	7,8	9,88	10,72	12,6	14,04	15,64	17,52	17,52	
1	3,04	4,6	6,36	8,08	10,44	11,8	13,6	15,28	16,92	18,88	1	5,04	8,52	12,32	16,16	20,4	23,56	27,52	31,2	34,8	38,72	38,72	
1	6,4	11,64	16,96	22,4	28,16	33,2	38,72	43,96	49,28	54,6	1	9,52	17,68	26,08	34,32	43,4	51,16	59,64	67,8	76,2	84,48	84,48	
1	3,2	5,68	8,28	10,44	13,24	15,16	17,44	19,92	22,52	24,6	1	3,84	6,64	9,64	12,2	15,6	18	20,76	23,56	26,56	29,2	29,2	
1	6,68	12	17,72	23,28	29,16	34	39,76	45,28	50,8	56,28	1	9,88	18,48	27,28	35,76	45,28	53,52	62,28	70,8	79,68	88,28	88,28	
1	1,28	2,36	3,28	3,96	4,64	4,8	5,6	6,4	7,6	7,76	1	3	3,28	5,08	6,48	8,48	9,28	10,68	12,48	12,76	14,76	14,76	
1	3	3,76	5,52	6,56	8,56	9,4	11,04	12,2	13,56	14,96	1	2,88	4,72	6,4	8,24	10,6	12,08	14	15,6	17,52	19,32	19,32	
1	1,48	2	3,12	3,44	4,52	4,72	5,28	6,16	6,92	7,16	1	1,04	1,6	1,68	1,76	2,4	2,4	2,52	2,56	3,16	3,16	3,16	
1	9,08	16,88	25,24	33,08	41,56	48,84	56,92	65,04	73,08	80,92	1	7	12,6	18,72	24,52	30,96	36,4	42,36	48,32	54,08	59,92	59,92	
1	1,24	1,84	3,24	3,92	4,16	4,2	5	6,28	7	7,12	1	3,16	4,84	7,16	8,96	11,56	13	15	17,12	19	20,96	20,96	
1	1,44	2,92	3,36	4,28	5,4	5,52	6,72	6,88	8,64	8,8	1	2,8	4,12	5,6	7,08	9,04	10	11,72	12,96	14,52	16,08	16,08	
1	1	1,04	2,6	2,6	3,04	3,04	3,04	4,6	4,64	4,64	1	2,84	4,08	5,4	7,04	9	10,04	11,8	13	14,36	16,08	16,08	
1	1	1,08	1,4	1,48	3,08	3,08	3,16	3,48	3,56	3,56	1	2,8	3,28	4,88	5,48	7,32	8,16	9,56	10,36	11,64	12,64	12,64	
1,0000	2,4521	4,0036	5,8721	7,3006	9,4242	10,5267	12,1000	13,8139	15,5067	16,8133	1,0000	3,0861	5,0964	7,2333	9,1709	11,7582	13,3400	15,4709	17,4139	19,6145	21,5097	21,5097	
1	2,4521	4,0036	5,8721	7,3006	9,4242	10,5267	12,1	13,8139	15,5067	16,8133	1	3,0861	5,0964	7,2333	9,1709	11,7582	13,34	15,4709	17,4139	19,6145	21,5097	21,5097	

● Número de nodos hoja

NumLeaves	Porcentajes de Consolidación	PCTBagging																					
		0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
		Balanceadas y 99% coverage										100 muestras Bootstrap											
		1	2	3	4	5	6	7	8	9	10	11											
Datasets		1	2	3	4	5	6	7	8	9	10	11											
abalone19	(50)	1	1,94	2,62	3,54	4,28	5,3	5,78	6,64	7,3	8,02	8,82	1	1	1	1	1	1	1	1	1	1	1
abalone9-18	(50)	1	1,34	1,58	2,04	2,18	3,16	3,24	3,58	3,72	4,14	4,28	1	1,96	2,76	3,66	4,56	5,72	6,4	7,44	8,32	9,22	10,22
ecoli-0	(50)	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2	2
ecoli1	(50)	1	1,58	1,76	2,34	2,52	3,42	3,68	4,1	4,44	4,86	5,2	1	2	3,14	4,18	5,26	6,54	7,3	8,38	9,42	10,56	11,56
ecoli2	(50)	1	1,02	1,84	2,02	2,1	2,86	2,86	2,96	3,12	3,96	3,96	1	1,8	2,26	2,96	3,54	4,5	4,88	5,64	6,16	6,8	7,42
ecoli3	(50)	1	1,02	1,34	1,98	2,12	2,36	2,36	2,52	3,14	3,48	3,48	1	1,94	2,88	3,78	4,78	5,8	6,4	7,4	8,3	9,24	10,18
ecoli4	(50)	1	1	1,5	1,66	1,68	2,5	2,5	2,52	2,68	3,18	3,18	1	1,06	1,72	2,06	2,18	2,78	2,8	2,96	3,26	3,96	3,98
glass-0-1-2-3	(50)	1	1,04	1,54	1,92	2,16	2,58	2,58	2,86	3,2	3,74	3,74	1	1,52	2,06	2,6	3,04	3,72	3,92	4,54	4,9	5,62	5,96
glass0	(50)	1	1,84	2,5	3,28	4,02	5,06	5,56	6,4	7,08	7,84	8,58	1	2,26	3,62	4,98	6,16	7,74	8,8	10,06	11,34	12,78	13,96
glass1	(50)	1	2,14	3,36	4,58	5,74	7,24	8,16	9,34	10,54	11,78	12,9	1	2,8	4,4	6,2	7,78	9,76	11,28	12,98	14,66	16,38	18,06
glass2	(50)	1	1,08	1,78	2,08	2,5	2,86	2,9	3,36	3,62	4,36	4,4	1	1,82	2,42	3,1	3,86	4,76	5,22	6,04	6,66	7,32	8,08
glass4	(50)	1	1	1,02	1,96	1,96	2,02	2,02	2,02	2,96	2,98	2,98	1	1,68	1,98	2,68	2,9	3,68	3,9	4,56	4,82	5,56	5,8
glass5	(50)	1	1	1	1,2	1,2	2	2	2	2,2	2,2	2,2	1	1	2	2	2,18	3	3	3,18	3,18	4,18	4,18
glass6	(50)	1	1	1,24	1,76	1,78	2,24	2,24	2,26	2,78	3,02	3,02	1	1,02	1,56	1,92	1,94	2,58	2,58	2,62	2,96	3,52	3,52
haberman(nlb)	(50)	1	1,76	2,5	3,2	3,84	4,88	5,32	6,04	6,66	7,48	8,16	1	2,34	3,68	5,14	6,48	7,94	9,02	10,42	11,82	13,22	14,5
ins0	(50)	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	2	2	2	2	2	2	2
new-thyroid1	(50)	1	1	1,06	1,92	1,92	2,06	2,06	2,06	2,92	2,98	2,98	1	1,36	1,84	2,36	2,72	3,22	3,28	3,92	4,16	4,92	5
new-thyroid2	(50)	1	1	1,08	1,94	1,94	2,08	2,08	2,08	2,94	3,02	3,02	1	1,22	1,8	2,22	2,62	3,02	3,04	3,64	3,86	4,64	4,66
page-blocks0	(50)	1	3,02	4,94	6,9	8,8	10,98	12,7	14,66	16,56	18,54	20,5	1	5,28	9,44	13,74	17,9	22,36	26,28	30,62	34,74	39,08	43,18
svm(nlb)	(50)	1	4,36	7,66	10,98	14,24	17,84	20,96	24,34	27,54	30,96	34,2	1	6,72	12,38	18,08	23,66	29,56	34,98	40,68			

50% Atributos																					PCT
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balanceadas y 99% coverage											100 muestras modo RSM										
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	1,94	2,78	3,78	4,62	5,66	6,04	7,02	7,7	8,46	9,22	1	1	1	1	1	1	1	1	1	1	1
1	1,32	1,54	2,14	2,26	3,26	3,34	3,62	3,78	4,18	4,32	1	1,82	2,62	3,4	4,12	5,18	5,7	6,62	7,18	7,98	8,7
1	1,02	1,14	1,16	1,24	2,16	2,16	2,26	2,26	2,4	2,4	1	1	1	1	2	2	2	2	2	2	2
1	1,4	1,6	2,02	2,18	3,1	3,22	3,56	3,8	4,18	4,4	1	1,52	1,84	2,6	2,9	3,48	3,72	4,2	4,78	5,28	5,62
1	1,08	1,8	2,08	2,2	2,88	2,88	3,08	3,28	4,08	4,08	1	1,64	2,06	2,64	3,06	3,78	4,04	4,76	5,04	5,76	6,1
1	1	1,46	1,96	2,1	2,46	2,46	2,6	3,1	3,56	3,56	1	1,66	2,46	3,14	3,96	4,72	5,02	5,88	6,52	7,36	7,98
1	1	1,52	1,6	1,72	2,52	2,52	2,64	2,72	3,24	3,24	1	1	1,46	2	2,08	2,46	2,46	2,54	3,08	3,54	3,54
1	1,11	1,58	1,94	2,16	2,68	2,72	3	3,3	3,84	3,88	1	1,66	1,96	2,66	3,06	3,78	4,18	4,68	5,28	5,68	6,24
1	1,96	2,72	3,6	4,5	5,56	6,04	6,96	7,82	8,6	9,54	1	2	2,94	4,1	5,06	6,2	6,9	7,94	9,02	10,04	10,96
1	2,24	3,5	4,78	6	7,42	8,44	9,78	10,94	12,32	13,44	1	1,98	2,84	3,82	4,72	5,94	6,68	7,68	8,56	9,52	10,4
1	1,06	1,82	2,06	2,62	2,88	2,88	3,5	3,68	4,5	4,5	1	1,74	2,28	2,8	3,52	4,36	4,72	5,46	5,96	6,52	7,24
1	1	1,02	1,9	1,9	2,02	2,02	2,02	2,9	2,92	2,92	1	1,28	2	2,28	2,8	3,28	3,34	4,08	4,14	5,08	5,14
1	1	1	1,14	1,14	2	2	2	2,14	2,14	2,14	1	1	2	2	2	3	3	3	3	4	4
1	1,02	1,22	1,68	1,7	2,24	2,24	2,28	2,72	2,94	2,94	1	1,1	1,54	2,08	2,16	2,64	2,64	2,82	3,26	3,8	3,8
1	1,74	2,32	3,02	3,62	4,64	4,88	5,66	6,18	6,94	7,5	1	1,04	1,26	1,56	1,62	2,06	2,08	2,16	2,44	2,68	2,7
1	1	1,08	1,08	1,1	2,08	2,08	2,1	2,1	2,18	2,18	1	1	1	1	2	2	2	2	2	2	2
1	1	1,18	1,94	1,94	2,18	2,18	2,18	2,94	3,12	3,12	1	1,2	1,72	2,2	2,68	2,92	2,92	3,6	3,88	4,6	4,6
1	1	1,18	1,94	1,98	2,18	2,18	2,22	2,98	3,16	3,16	1	1,22	1,64	2,22	2,5	2,86	2,86	3,36	3,72	4,36	4,36
1	2,98	5,08	7,02	9,06	11,3	13,12	15,16	17,1	19,2	21,18	1	4,7	8,24	11,96	15,52	19,48	22,92	26,6	30,2	33,86	37,44
1	3,94	6,72	9,62	12,38	15,5	18,2	21,06	23,86	26,74	29,58	1	2,96	4,94	6,92	8,9	11,08	12,76	14,8	16,72	18,76	20,66
1	1,46	2,1	2,5	3,34	3,8	4	4,88	5,24	5,92	6,34	1	2	2,74	3,52	4,4	5,44	5,86	6,8	7,52	8,32	9,26
1	2,06	2,94	3,84	4,84	5,98	6,72	7,74	8,62	9,52	10,56	1	3,02	4,8	6,7	8,68	10,8	12,3	14,32	16,18	18	19,98
1	3,82	6,44	9,3	11,94	15,06	17,64	20,36	23,14	25,84	28,58	1	5,18	9,32	13,48	17,54	22	25,94	30,1	34,16	38,4	42,48
1	2,08	3,38	4,54	5,66	7,14	8,12	9,3	10,4	11,76	12,78	1	2,42	3,84	5,32	6,58	8,28	9,5	10,9	12,24	13,8	15,08
1	3,82	6,38	9,2	11,88	14,84	17,32	20,1	22,82	25,48	28,2	1	5,5	9,88	14,38	18,76	23,48	27,68	32,22	36,56	41,1	45,44
1	1,08	1,64	2,08	2,36	2,72	2,78	3,08	3,5	4,08	4,14	1	2	2,12	3,04	3,74	4,74	5,12	5,82	6,74	6,86	7,86
1	2	2,64	3,16	3,58	4,62	4,88	5,76	6,2	6,92	7,46	1	1,94	2,88	3,72	4,64	5,8	6,52	7,48	8,28	9,26	10,16
1	1,28	1,42	1,92	1,98	2,7	2,82	3,04	3,38	3,68	3,8	1	1,02	1,3	1,34	1,38	1,7	1,7	1,76	1,78	2,08	2,08
1	4,88	8,56	12,34	16,08	20,08	23,62	27,48	31,14	34,94	38,7	1	3,88	6,62	9,56	12,38	15,5	18,1	21,02	23,86	26,7	29,48
1	1,18	1,4	2,18	2,4	2,58	2,64	2,98	3,64	3,98	4,04	1	2,08	2,96	4,06	5	6,34	7,04	8,08	9,08	10,06	11,04
1	1,24	1,98	2,2	2,62	3,24	3,34	3,88	3,98	4,84	4,96	1	1,92	2,56	3,32	4	5,02	5,52	6,38	6,96	7,78	8,52
1	1	1,14	1,72	1,78	2,14	2,14	2,2	2,78	2,92	2,92	1	1,92	2,52	3,16	3,98	4,96	5,46	6,34	6,92	7,58	8,44
1	1,02	1,04	1,2	1,2	2,06	2,08	2,08	2,24	2,26	2,28	1	1,86	2,06	2,88	3,18	4,04	4,34	5,08	5,46	6,1	6,52
1,0000	1,7188	2,5133	3,4133	4,1236	5,2024	5,7485	6,5327	7,3448	8,2073	8,8503	1,0000	2,0382	3,0424	4,1170	5,0885	6,3733	7,1521	8,2267	9,1976	10,2988	11,2370
<b>1</b>	<b>1,7188</b>	<b>2,5133</b>	<b>3,4133</b>	<b>4,1236</b>	<b>5,2024</b>	<b>5,7485</b>	<b>6,5327</b>	<b>7,3448</b>	<b>8,2073</b>	<b>8,8503</b>	<b>1</b>	<b>2,0382</b>	<b>3,0424</b>	<b>4,117</b>	<b>5,0885</b>	<b>6,3733</b>	<b>7,1521</b>	<b>8,2267</b>	<b>9,1976</b>	<b>10,2988</b>	<b>11,237</b>

75% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balanceadas y 99% coverage											100 muestras modo RSM										
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
1	1,92	2,6	3,46	4,2	5,18	5,6	6,5	7,08	7,78	8,52	1	1	1	1	1	1	1	1	1	1	1
1	1,32	1,64	2,08	2,22	3,22	3,32	3,62	3,8	4,28	4,44	1	1,82	2,58	3,32	4,02	5,08	5,68	6,5	7,12	7,9	8,68
1	1,02	1,06	1,08	1,12	2,08	2,08	2,14	2,14	2,2	2,2	1	1	1	1	1	2	2	2	2	2	2
1	1,5	1,74	2,2	2,46	3,32	3,54	4	4,26	4,7	5	1	1,52	1,84	2,6	2,9	3,48	3,72	4,2	4,78	5,28	5,62
1	1,08	1,8	2,08	2,24	2,88	2,88	3,12	3,32	4,12	4,12	1	1,64	2,06	2,64	3,06	3,78	4,04	4,76	5,04	5,76	6,1
1	1	1,38	1,96	2,16	2,38	2,38	2,58	3,16	3,54	3,54	1	1,68	2,54	3,18	4	4,84	5,28	6,14	6,74	7,64	8,28
1	1	1,54	1,66	1,78	2,54	2,54	2,66	2,78	3,32	3,32	1	1	1,46	2	2,06	2,46	2,46	2,52	3,06	3,52	3,52
1	1,04	1,56	1,88	2,14	2,6	2,6	2,9	3,18	3,74	3,74	1	1,64	1,96	2,64	3,02	3,74	4,16	4,62	5,22	5,62	6,18
1	1,92	2,58	3,52	4,34	5,32	5,94	6,8	7,7	8,4	9,28	1	2	2,94	4,1	5,04	6,2	6,96	7,98	9,06	10,08	11
1	2,22	3,32	4,7	5,84	7,2	8,12	9,34	10,64	11,82	12,96	1	1,94	2,74	3,74	4,62	5,82	6,5	7,46	8,38	9,26	10,12
1	1,08	1,82	2,08	2,52	2,9	2,9	3,42	3,6	4,42	4,42	1	1,66	2,24	2,68	3,36	4,14	4,52	5,22	5,64	6,24	6,88
1	1	1,02	1,94	1,94	2,02	2,02	2,02	2,94	2,96	2,96	1	1,28	2	2,28	2,86	3,28	3,36	4,14	4,22	5,14	5,22
1	1	1	1,16	1,16	2	2	2	2,16	2,16	2,16	1	1	2	2	2	3	3	3	3	4	4
1	1	1,14	1,72	1,72	2,14	2,14	2,14	2,72	2,86	2,86	1	1,08	1,48	2,06	2,16	2,56	2,56	2,74	3,24	3,72	3,72
1	1,74	2,32	3,02	3,62	4,64	4,88	5,66	6,18	6,94	7,5	1	1,04	1,26	1,56	1,62	2,06	2,08	2,16	2,44	2,68	2,7
1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2	2
1	1	1,08	1,94	1,94	2,08	2,08	2,08	2,94	3,02	3,02	1	1,26	1,78	2,26	2,7	3,04	3,04	3,74	3,96	4,74	4,74
1	1	1,1	1,9	1,9	2,1	2,1	2,1	2,9	3	3	1	1,24	1,74	2,24	2,56	2,98	2,98	3,54	3,8	4,54	4,54
1	3,08	5,02	6,98	8,96	11,26	13	15,06	16,94	18,96	20,96	1	4,66	8,22	11,86	15,42	19,36	22,74	26,4	29,94	33,6	37,16
1	4,22	7,24	10,42	13,48	16,9	19,84	23	26,08	29,2	32,32	1	3,06	5,14	7,1	9,18	11,44	13,14	15,3	17,18	19,34	21,32
1	1,42	2,12	2,42	3,32	3,74	3,9	4,86	5,1	5,86	6,22	1	2	2,74	3,52	4,4	5,44	5,86	6,8	7,52	8,32	9,26
1	2	2,7	3,6	4,44	5,54	6,14	7,02	7,88	8,62	9,58	1	3,02	4,76	6,66	8,58	10,7	12,28	14,26	16,1	17,9	19,86
1	3,76	6,46	9,22	11,8	14,86	17,4	20,06	22,74	25,52	28,2	1	5,26	9,34	13,52	17,66	22,18	26,06	30,3	34,38	38,56	42,72
1	2,04	3,34	4,6	5,74	7,02	7,98	9,12	10,38	11,68	12,72	1	2,42	3,82	5,32	6,6	8,3	9				

50% Atributos																					
0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%	0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%
Balances y 99% coverage											100 muestras modo RSM										
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
1	1,92	2,68	3,66	4,42	5,5	5,96	6,86	7,54	8,3	9,04	1	1	1	1	1	1	1	1	1	1	1
1	1,22	1,38	1,74	1,82	2,78	2,84	3,02	3,18	3,46	3,58	1	1,84	2,64	3,38	4,1	5,16	5,68	6,6	7,18	7,92	8,7
1	1	1,06	1,06	1,06	2,06	2,06	2,06	2,06	2,12	2,12	1	1	1	1	1	2	2	2	2	2	2
1	1,52	1,76	2,32	2,5	3,34	3,52	3,98	4,26	4,78	5,02	1	1,52	1,84	2,6	2,9	3,48	3,72	4,2	4,78	5,28	5,62
1	1,08	1,78	2,08	2,26	2,86	2,88	3,12	3,36	4,12	4,14	1	1,64	2,06	2,64	3,06	3,78	4,04	4,76	5,04	5,76	6,1
1	1	1,4	1,94	2,1	2,4	2,4	2,56	3,1	3,5	3,5	1	1,68	2,5	3,16	3,96	4,74	5,1	5,94	6,56	7,42	8,06
1	1	1,54	1,68	1,74	2,54	2,54	2,6	2,74	3,28	3,28	1	1	1,46	2	2,06	2,46	2,46	2,52	3,06	3,52	3,52
1	1,08	1,54	1,92	2,14	2,62	2,66	2,92	3,26	3,76	3,8	1	1,66	1,96	2,66	3,06	3,78	4,18	4,68	5,28	5,68	6,24
1	1,98	2,56	3,52	4,24	5,34	5,96	6,78	7,64	8,32	9,2	1	2,02	2,98	4,12	5,06	6,24	7	8,04	9,08	10,14	11,06
1	2,12	3,34	4,54	5,7	7,04	7,94	9,16	10,3	11,58	12,64	1	1,98	2,86	3,86	4,76	5,98	6,76	7,74	8,66	9,62	10,52
1	1,1	1,8	2,1	2,56	2,9	2,9	3,46	3,66	4,46	4,46	1	1,7	2,24	2,74	3,46	4,26	4,58	5,32	5,8	6,36	7,04
1	1	1	1,9	1,9	2	2	2	2,9	2,9	2,9	1	1,24	2	2,24	2,84	3,24	3,28	4,08	4,12	5,08	5,12
1	1	1	1,16	1,16	2	2	2	2,16	2,16	2,16	1	1	2	2	2	3	3	3	3	4	4
1	1	1,2	1,6	1,6	2,2	2,2	2,2	2,6	2,8	2,8	1	1,1	1,52	2,08	2,16	2,62	2,62	2,8	3,26	3,78	3,78
1	1,7	2,28	3,02	3,52	4,54	4,92	5,54	6,16	6,86	7,44	1	1,04	1,26	1,56	1,62	2,06	2,08	2,16	2,44	2,68	2,7
1	1	1,06	1,06	1,06	2,06	2,06	2,06	2,06	2,12	2,12	1	1	1	1	1	2	2	2	2	2	2
1	1	1,2	1,94	1,94	2,2	2,2	2,2	2,94	3,14	3,14	1	1,16	1,74	2,16	2,64	2,9	2,9	3,54	3,8	4,54	4,54
1	1	1,08	1,94	1,94	2,08	2,08	2,08	2,94	3,02	3,02	1	1,18	1,7	2,18	2,54	2,88	2,88	3,42	3,72	4,42	4,42
1	3,14	5,22	7,34	9,36	11,74	13,6	15,68	17,74	19,88	21,96	1	4,64	8,22	11,9	15,5	19,44	22,76	26,48	30,04	33,74	37,26
1	3,98	6,86	9,76	12,64	15,8	18,5	21,44	24,28	27,22	30,14	1	2,94	4,94	6,82	8,8	10,94	12,54	14,58	16,4	18,46	20,34
1	1,44	2,16	2,48	3,4	3,84	4,04	4,98	5,28	6,02	6,44	1	2	2,74	3,52	4,4	5,44	5,86	6,8	7,52	8,32	9,26
1	2,02	2,7	3,7	4,52	5,72	6,36	7,24	8,18	8,92	9,88	1	3,02	4,76	6,66	8,6	10,72	12,26	14,26	16,1	17,9	19,86
1	3,74	6,3	9,1	11,68	14,58	17,1	19,76	22,48	25,12	27,78	1	5,32	9,46	13,82	17,92	22,5	26,54	30,76	35	39,26	43,46
1	2,02	3,28	4,52	5,62	7,06	7,96	9,08	10,3	11,58	12,58	1	2,42	3,8	5,32	6,62	8,3	9,48	10,88	12,3	13,78	15,1
1	3,7	6,22	8,92	11,48	14,34	16,76	19,42	22,02	24,64	27,24	1	5,46	9,84	14,2	18,5	23,16	27,36	31,8	36,02	40,54	44,86
1	1,12	1,64	2,12	2,32	2,76	2,78	3,08	3,46	4,08	4,1	1	2	2,1	3,04	3,74	4,74	5,1	5,8	6,74	6,84	7,84
1	1,98	2,26	3,12	3,52	4,52	4,86	5,7	6,12	6,84	7,38	1	1,94	2,86	3,7	4,62	5,8	6,54	7,5	8,3	9,26	10,16
1	1,26	1,54	2,04	2,18	2,8	2,9	3,2	3,54	3,98	4,08	1	1,04	1,3	1,36	1,42	1,72	1,72	1,8	1,84	2,12	2,14
1	4,7	8,32	12,02	15,66	19,62	22,98	26,68	30,32	34	37,64	1	3,94	6,76	9,68	12,54	15,72	18,48	21,4	24,26	27,14	30,02
1	1,18	1,46	2,2	2,48	2,7	2,74	3,14	3,76	4,16	4,22	1	2,06	2,96	4,04	4,96	6,28	7	8	9	9,98	10,96
1	1,26	1,98	2,22	2,6	3,28	3,32	3,88	3,94	4,84	4,92	1	1,9	2,54	3,28	4	4,98	5,46	6,32	6,92	7,7	8,46
1	1	1,14	1,74	1,76	2,14	2,14	2,16	2,76	2,9	2,9	1	1,92	2,54	3,2	4,02	5,02	5,56	6,42	7,04	7,7	8,58
1	1,02	1,02	1,26	1,26	2,04	2,04	2,06	2,28	2,3	2,3	1	1,9	2,14	2,94	3,24	4,14	4,5	5,24	5,6	6,28	6,74
1.0000	1,7055	2,4776	3,3855	4,0648	5,1333	5,6727	6,4273	7,2521	8,0958	8,7248	1,0000	2,0382	3,0521	4,1170	5,0939	6,3782	7,1648	8,2376	9,2079	10,3097	11,2564
<b>1</b>	<b>1,7055</b>	<b>2,4776</b>	<b>3,3855</b>	<b>4,0648</b>	<b>5,1333</b>	<b>5,6727</b>	<b>6,4273</b>	<b>7,2521</b>	<b>8,0958</b>	<b>8,7248</b>	<b>1</b>	<b>2,0382</b>	<b>3,0521</b>	<b>4,117</b>	<b>5,0939</b>	<b>6,3782</b>	<b>7,1648</b>	<b>8,2376</b>	<b>9,2079</b>	<b>10,3097</b>	<b>11,2564</b>

75% Atributos																					
0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%	0,00%	10,00%	20,00%	30,00%	40,00%	50,00%	60,00%	70,00%	80,00%	90,00%	100,00%
Balances y 99% coverage											100 muestras modo RSM										
78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
1	1,92	2,52	3,48	4,18	5,2	5,5	6,44	7,06	7,76	8,44	1	1	1	1	1	1	1	1	1	1	1
1	1,32	1,46	1,98	2,1	3,02	3,08	3,42	3,5	3,88	3,96	1	1,82	2,58	3,32	4	5,06	5,66	6,48	7,06	7,86	8,62
1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2	2
1	1,52	1,82	2,36	2,68	3,46	3,6	4,18	4,46	5,02	5,28	1	1,52	1,84	2,6	2,9	3,48	3,72	4,2	4,78	5,28	5,62
1	1,16	1,86	2,16	2,22	3,04	3,06	3,24	3,42	4,24	4,28	1	1,64	2,06	2,64	3,06	3,78	4,04	4,76	5,04	5,76	6,1
1	1	1,34	1,96	2,08	2,34	2,34	2,46	3,08	3,42	3,42	1	1,68	2,56	3,2	4,02	4,84	5,28	6,14	6,74	7,66	8,3
1	1	1,54	1,64	1,68	2,54	2,54	2,58	2,68	3,22	3,22	1	1	1,46	2	2,06	2,46	2,46	2,52	3,06	3,52	3,52
1	1,06	1,5	1,94	2,1	2,56	2,58	2,78	3,18	3,66	3,68	1	1,64	1,96	2,64	3,02	3,74	4,16	4,62	5,22	5,62	6,18
1	1,92	2,64	3,46	4,32	5,36	5,9	6,78	7,58	8,32	9,22	1	2	2,94	4,1	5,04	6,18	6,94	7,96	9,04	10,06	10,98
1	2,14	3,32	4,54	5,66	7,04	8,04	9,18	10,38	11,58	12,7	1	1,94	2,72	3,74	4,6	5,8	6,48	7,42	8,36	9,22	10,08
1	1,1	1,78	2,1	2,52	2,9	2,92	3,4	3,66	4,4	4,44	1	1,66	2,24	2,68	3,36	4,14	4,52	5,22	5,64	6,24	6,88
1	1	1,02	1,92	1,92	2,02	2,02	2,02	2,92	2,94	2,94	1	1,28	2	2,28	2,86	3,28	3,34	4,14	4,2	5,14	5,2
1	1	1	1,18	1,18	2	2	2	2,18	2,18	2,18	1	1	2	2	2	3	3	3	3	4	4
1	1	1,22	1,74	1,74	2,22	2,22	2,22	2,74	2,96	2,96	1	1,08	1,48	2,06	2,14	2,56	2,56	2,72	3,22	3,7	3,7
1	1,7	2,28	3,02	3,52	4,54	4,92	5,54	6,16	6,86	7,44	1	1,04	1,26	1,56	1,62	2,06	2,08	2,16	2,44	2,68	2,7
1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	1	2	2	2	2	2	2
1	1	1,14	1,9	1,92	2,14	2,14	2,16	2,92	3,06	3,06	1	1,24	1,76	2,24	2,68	3	3	3,68	3,92	4,68	4,68
1	1	1,06	1,9	1,9	2,06	2,06	2,06	2,9	2,96	2,96	1	1,24	1,74	2,24	2,56	2,98	2,98	3,54	3,8	4,54	4,54
1	3,02	4,96	6,98	8,96	11,22	13,02	15,02	17,02	18,98	20,98	1	4,68	8,24	11,9	15,44	19,42	22,8	26,46	30	33,68	37,24
1	4,22	7,14	10,36	13,4	16,78	19,62	22,82	25,88	28,96	32,02	1	3,06	5,14	7,1	9,18	11,44	13,14	15,3	17,18	19,34	21,32
1	1,46	2,14	2,46	3,36	3,82	4	4,96	5,22	5,96	6,36	1	2	2,74	3,52	4,4	5,44	5,86	6,8	7,52	8,32	9,26
1	2,02	2,8	3,68	4,54	5,72	6,4	7,3	8,14	8,96	9,94	1	3,02	4,76	6,66	8,58	10,7	12,28	14,26	16,1	17,9	19,86
1	3,7	6,32	8,98	11,7	14,58	17,1	19,86	22,48	25,14	27,8	1	5,26	9,34	13,54	17,66	22,2	26,08	30,32	34,4	38,6	42,74
1	2,1	3,34	4,64	5,72	7,12	8,08	9,22	10,46	11,76	12,8	1	2,42	3,82	5,32	6,6	8,3	9,5	10			

# • Tiempo de construcción

Time		PCTBagging																						
		Balaceadas y 99% coverage										100 muestras Bootsrap												
Porcentajes de Consolidación		0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %	0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %	
Datasets		1	2	3	4	5	6	7	8	9	10	11												
abalone19 (50)		0.3513	0.3491	0.3344	0.2906	0.2825	0.2781	0.2791	0.2803	0.2844	0.2747	0.2809	1.4986	1.493	1.5054	1.5136	1.5064	1.6406	1.4498	1.588	1.512	1.5414	1.5366	
abalone9-18 (50)		0.0359	0.0363	0.0372	0.0338	0.0337	0.0341	0.0322	0.035	0.035	0.0328	0.035	0.288	0.261	0.2508	0.2458	0.2392	0.2606	0.2282	0.2482	0.2322	0.239	0.2424	
ecoli-0 (50)		0.0025	0.0019	0.0028	0.0012	0.0025	0.0025	0.0031	0.0053	0.0047	0.0041	0.0047	0.0274	0.0286	0.0278	0.0278	0.0204	0.019	0.0208	0.0198	0.0192	0.0196	0.0196	
ecoli1 (50)		0.0069	0.0062	0.0072	0.0069	0.0059	0.0062	0.0066	0.0084	0.0087	0.0094	0.0128	0.08	0.068	0.0718	0.0624	0.0614	0.0654	0.0562	0.0614	0.0578	0.0596	0.0588	
ecoli2 (50)		0.0094	0.01	0.0097	0.01	0.0094	0.0094	0.0103	0.0122	0.0116	0.0109	0.0172	0.0878	0.083	0.0772	0.0752	0.0716	0.0784	0.068	0.0734	0.0682	0.0688	0.068	
ecoli3 (50)		0.0112	0.0112	0.0103	0.0088	0.0084	0.0103	0.0112	0.0119	0.0109	0.01	0.0166	0.0848	0.0744	0.0664	0.0608	0.059	0.064	0.0554	0.0592	0.058	0.0544	0.0544	
ecoli4 (50)		0.0069	0.0094	0.0091	0.0069	0.0081	0.0072	0.0081	0.0087	0.0097	0.0087	0.0131	0.0572	0.058	0.0506	0.0478	0.0466	0.0504	0.0434	0.0492	0.0462	0.044	0.044	
glass-0-1-2-3 (50)		0.0044	0.0069	0.0075	0.0047	0.0053	0.0059	0.005	0.0063	0.0075	0.0062	0.0091	0.068	0.0586	0.0526	0.0486	0.047	0.0508	0.045	0.048	0.0446	0.044	0.0436	
glass0 (50)		0.0072	0.0063	0.0131	0.0066	0.0062	0.0066	0.0056	0.0069	0.0075	0.0075	0.0091	0.114	0.0994	0.0898	0.0828	0.0758	0.0784	0.0672	0.0728	0.065	0.0652	0.0716	
glass1 (50)		0.0072	0.0081	0.0094	0.0053	0.0059	0.0066	0.0075	0.0081	0.0078	0.0081	0.0094	0.1356	0.117	0.1066	0.099	0.0934	0.098	0.0874	0.0894	0.0812	0.0844	0.0816	
glass2 (50)		0.0116	0.0122	0.0119	0.0097	0.0103	0.0106	0.0097	0.01	0.0119	0.0112	0.0131	0.0758	0.073	0.067	0.0572	0.0534	0.0554	0.0556	0.0556	0.0486	0.0502	0.0508	
glass4 (50)		0.0069	0.0078	0.0094	0.0075	0.0059	0.0084	0.0069	0.0084	0.0084	0.0097	0.0087	0.0522	0.0536	0.041	0.0334	0.031	0.0332	0.0332	0.031	0.031	0.0282	0.0312	
glass5 (50)		0.0053	0.0059	0.0075	0.0062	0.0066	0.0059	0.0066	0.0072	0.0069	0.0066	0.0069	0.0348	0.0444	0.0202	0.0194	0.02	0.0198	0.0192	0.0186	0.0178	0.017	0.0194	
glass6 (50)		0.0075	0.0059	0.0069	0.0056	0.0069	0.0059	0.0063	0.0069	0.0081	0.0062	0.0069	0.0602	0.0992	0.0504	0.0458	0.0442	0.0464	0.041	0.0448	0.0434	0.0418	0.0484	
haberman10 (50)		0.0056	0.0056	0.0059	0.0059	0.005	0.0062	0.0078	0.0081	0.0081	0.0081	0.0103	0.1146	0.0926	0.0862	0.0792	0.0772	0.0836	0.075	0.083	0.0894	0.0746	0.0808	
iris0 (50)		0.0013	0.0006	0.0012	0.0012	0.0016	0.0009	0.0025	0.0031	0.0028	0.0041	0.0047	0.0092	0.009	0.0102	0.0092	0.009	0.0056	0.0052	0.0066	0.0046	0.0056	0.0062	
new-thyroid1 (50)		0.0044	0.0037	0.0041	0.0034	0.0028	0.0034	0.0047	0.005	0.0053	0.0044	0.0069	0.035	0.0328	0.0306	0.0262	0.0252	0.025	0.0222	0.0232	0.025	0.0226	0.0222	
new-thyroid2 (50)		0.0037	0.0044	0.0047	0.0028	0.0031	0.0031	0.0037	0.0053	0.0047	0.0047	0.0059	0.0342	0.035	0.031	0.0266	0.0246	0.025	0.023	0.0248	0.0236	0.0226	0.0218	
page-blocks0 (50)		0.4547	0.4006	0.3662	0.3244	0.3056	0.3109	0.3006	0.3106	0.3003	0.2969	0.3059	7.2586	5.2292	4.437	4.2714	4.3152	4.647	4.3884	4.4864	4.5356	4.4286	4.3782	
svm10 (50)		0.0334	0.0287	0.0241	0.0244	0.0247	0.0256	0.025	0.0237	0.0247	0.0256	0.0263	0.6044	0.4586	0.4112	0.3874	0.3794	0.4042	0.3832	0.3816	0.3766	0.3584	0.3516	
segment0 (50)		0.1616	0.1566	0.1316	0.1322	0.1153	0.1191	0.1153	0.1094	0.1094	0.1125	0.1147	1.9972	1.7192	1.6028	1.4324	1.5338	1.4584	1.3982	1.4172	1.3682	1.3682	1.3138	
ehc1c0 (50)		0.0569	0.0503	0.0456	0.0444	0.0413	0.0469	0.0391	0.0403	0.0397	0.0406	0.0416	0.6218	0.4494	0.4208	0.377	0.388	0.3698	0.3514	0.3528	0.3386	0.3272	0.3214	
ehc1e2 (50)		0.0928	0.0828	0.0744	0.0712	0.0675	0.0675	0.0697	0.065	0.0647	0.0659	0.0675	1.3246	0.972	0.8806	0.8242	0.854	0.8138	0.79	0.7814	0.739	0.7334	0.716	
ehc1e3 (50)		0.0569	0.0525	0.0459	0.0437	0.0409	0.0397	0.0378	0.0381	0.0381	0.0397	0.0403	0.6584	0.5248	0.479	0.4308	0.4356	0.4252	0.4014	0.41	0.3936	0.3796	0.3678	
ehc1e4 (50)		0.1041	0.0887	0.0775	0.0759	0.0725	0.0716	0.0684	0.0691	0.0684	0.0719	0.0706	1.3704	1.0448	0.916	0.8482	0.8768	0.8298	0.8126	0.78	0.7872	0.783	0.7438	
owel0 (50)		0.0503	0.0506	0.0431	0.0406	0.04	0.0422	0.0391	0.0403	0.0372	0.0384	0.0419	0.4812	0.4122	0.3714	0.3496	0.37	0.345	0.347	0.3378	0.3364	0.336	0.3244	
wisconsin10 (50)		0.0128	0.0125	0.0103	0.0122	0.0112	0.0125	0.01	0.0128	0.0122	0.0125	0.0138	0.1896	0.1704	0.1536	0.1404	0.1456	0.1394	0.137	0.1326	0.1316	0.1244	0.1268	
yeast-2 (50)		0.0225	0.0228	0.0209	0.0184	0.0203	0.02	0.02	0.0237	0.0181	0.02	0.0203	0.1496	0.151	0.1494	0.1394	0.154	0.1384	0.135	0.1316	0.1392	0.1348	0.1272	
yeast3 (50)		0.0753	0.0669	0.0563	0.0619	0.06	0.0616	0.0612	0.0594	0.0594	0.0597	0.0634	1.3616	0.9984	0.9068	0.8738	0.9174	0.8822	0.8668	0.8538	0.8556	0.835	0.8082	
yeast4 (50)		0.0588	0.0622	0.0559	0.0537	0.0513	0.0528	0.0512	0.055	0.0503	0.0544	0.055	0.5154	0.4398	0.4162	0.4362	0.4128	0.414	0.4442	0.399	0.4014	0.3858	0.3858	
yeast5 (50)		0.0694	0.0691	0.0619	0.0634	0.0584	0.0619	0.0631	0.0619	0.0597	0.0625	0.0609	0.4136	0.3648	0.345	0.3288	0.357	0.355	0.3474	0.3624	0.3326	0.336	0.3234	
yeast6 (50)		0.0469	0.0459	0.0422	0.0362	0.0406	0.0387	0.0391	0.0413	0.0378	0.0422	0.0394	0.242	0.1776	0.1736	0.1712	0.1826	0.1766	0.1732	0.1662	0.1654	0.1634	0.1736	
yeast7 (50)		0.07	0.0763	0.0656	0.0659	0.0662	0.0609	0.0631	0.0666	0.0594	0.0663	0.0675	0.459	0.3712	0.361	0.3574	0.3762	0.3454	0.3684	0.356	0.353	0.3736	0.4048	
Average		0.0562	0.0533	0.0489	0.0450	0.0432	0.0437	0.0430	0.0441	0.0431	0.0436	0.0455	0.6214	0.4928	0.4442	0.4215	0.4314	0.4377	0.4154	0.4240	0.4158	0.4111	0.4051	

Time		PCT																					
		50% Atributos										100 muestras modo RSM											
Porcentajes de Consolidación		0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %	0.00 %	10.00 %	20.00 %	30.00 %	40.00 %	50.00 %	60.00 %	70.00 %	80.00 %	90.00 %	100.00 %
Datasets		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
0.2628	0.2906	0.2847	0.2813	0.2694	0.2741	0.2728	0.2703	0.2638	0.2809	0.2697	0.6997	0.7	0.7031	0.6897	0.6813	0.6841	0.6931	0.7053	0.6803	0.6959	0.6928	0.6928	
0.0288	0.0306	0.03	0.0306	0.0312	0.0272	0.0322	0.0303	0.0291	0.0325	0.0309	0.1628	0.1431	0.1372	0.1331	0.1312	0.1309	0.1309	0.1294	0.1278	0.1325	0.13	0.13	
0.0069	0.0075	0.0069	0.0075	0.0081	0.0091	0.0072	0.0091	0.0097	0.0094	0.0094	0.0269	0.0253	0.0241	0.0238	0.0247	0.0147	0.0147	0.0156	0.0131	0.0147	0.015	0.015	
0.0113	0.0103	0.0116	0.0112	0.0116	0.0119	0.0122	0.0122	0.0125	0.0131	0.0081	0.0578	0.045	0.0431	0.04	0.0384	0.04	0.0387	0.0375	0.0359	0.0403	0.0369	0.0369	
0.0134	0.0119	0.0106	0.0119	0.0131	0.0109	0.0153	0.0128	0.0147	0.0147	0.0134	0.0591	0.0481	0.0416	0.0384	0.0378	0.0381	0.0362	0.0359	0.0337	0.0338	0.0331	0.0331	
0.0119	0.0134	0.0134	0.0122	0.0128	0.0137	0.0144	0.0153	0.0144	0.0144	0.0131	0.0563	0.0462	0.0419	0.0406	0.0381	0.0366	0.0409	0.0369	0.0362	0.0378	0.0375	0.0375	
0.0131	0.0116	0.0112	0.0122	0.0112	0.0109	0.0125	0.0125	0.0141	0.0128	0.0094	0.0491	0.0466	0.0394	0.035	0.035	0.0366	0.0347	0.0337	0.0325	0.0334	0.0328	0.0328	
0.0078	0.0081	0.0081	0.0081	0.01	0.0091	0.01	0.0109	0.0109	0.0125	0.0106	0.0628	0.0453	0.0419	0.0359	0.0362	0.0347	0.0341	0.0334	0.0303	0.0334	0.0306	0.0306	
0.0078	0.0084	0.0097	0.0112	0.01	0.0087	0.01	0.0097	0.0103	0.0106	0.0128	0.0634	0.0631	0.0584	0.0519	0.0494	0.0462	0.0469	0.0475	0.0437	0.0438	0.0416	0.0416	
0.0081	0.0097	0.0097	0.0112	0.0116	0.0094	0.0122	0.0112	0.0097	0.0112	0.0116	0.0931	0.0734	0.065										

RSM

75% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balanceadas y 99% coverage											100 muestras modo RSM										
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
0.3513	0.3334	0.3294	0.3125	0.3081	0.3078	0.2966	0.2953	0.2916	0.2916	0.2941	1.0116	1.0172	1.0225	1.0109	1.0069	0.9947	1.0119	1.0188	1.0206	1.0091	0.9831
0.0328	0.0328	0.0325	0.0319	0.0319	0.0294	0.0294	0.0294	0.0284	0.0294	0.0272	0.2378	0.2141	0.1959	0.1944	0.1938	0.1859	0.2019	0.19	0.1894	0.1881	0.1859
0.0019	0.0019	0.0016	0.0019	0.0016	0.0022	0.0019	0.0025	0.0022	0.0022	0.0016	0.0287	0.0272	0.0287	0.0275	0.0275	0.0172	0.0166	0.0178	0.0169	0.0188	0.0159
0.0062	0.005	0.0059	0.005	0.0041	0.0044	0.0056	0.005	0.0063	0.0066	0.0059	0.0644	0.0544	0.0522	0.0447	0.0459	0.0456	0.0447	0.045	0.0447	0.0416	0.0409
0.0094	0.0081	0.0078	0.0069	0.0066	0.0075	0.0075	0.0078	0.0078	0.0069	0.0084	0.0656	0.0547	0.0475	0.0469	0.0466	0.0422	0.0412	0.0434	0.0406	0.0419	0.04
0.0097	0.0078	0.0087	0.0081	0.0084	0.0069	0.0069	0.0069	0.0066	0.0078	0.0091	0.0681	0.0609	0.0569	0.0525	0.0503	0.0475	0.05	0.0484	0.0475	0.0469	0.0459
0.0075	0.0072	0.0062	0.0059	0.0069	0.005	0.0041	0.0059	0.0059	0.0059	0.0075	0.0541	0.0541	0.0475	0.0416	0.0409	0.0375	0.0409	0.04	0.0394	0.0406	0.0394
0.0047	0.005	0.005	0.005	0.0044	0.0053	0.0041	0.0037	0.005	0.005	0.0056	0.0706	0.0591	0.0522	0.0481	0.0478	0.0438	0.0456	0.0422	0.0397	0.0422	0.0406
0.0059	0.0053	0.0056	0.0059	0.0047	0.005	0.0041	0.0053	0.0047	0.0056	0.0072	0.1009	0.0834	0.0741	0.0737	0.0675	0.0634	0.06	0.0578	0.055	0.0569	0.0544
0.0081	0.0066	0.0059	0.0047	0.0059	0.0047	0.005	0.0047	0.0059	0.0063	0.0053	0.1159	0.0978	0.0875	0.0806	0.0781	0.0703	0.0712	0.0669	0.0669	0.0656	0.0647
0.0097	0.0103	0.0088	0.0103	0.0078	0.0094	0.0084	0.0084	0.0075	0.0087	0.0084	0.0669	0.0581	0.0519	0.0497	0.0484	0.0441	0.0425	0.04	0.0409	0.0413	0.0394
0.0081	0.0081	0.0063	0.0059	0.0066	0.0056	0.0081	0.0062	0.0059	0.0059	0.0072	0.0441	0.0416	0.0322	0.0297	0.0284	0.0253	0.0256	0.0234	0.0259	0.0247	0.0247
0.0059	0.0063	0.0072	0.0047	0.0056	0.0053	0.0047	0.0059	0.0053	0.0053	0.0059	0.0316	0.03	0.0175	0.0187	0.0188	0.0159	0.0166	0.0169	0.0181	0.0153	0.0153
0.0062	0.0059	0.0041	0.0044	0.005	0.0053	0.005	0.0041	0.0053	0.005	0.0059	0.0572	0.0553	0.0497	0.0434	0.0441	0.0419	0.0419	0.0437	0.0416	0.04	0.0403
0.0056	0.0038	0.005	0.0044	0.0028	0.0047	0.0037	0.0038	0.005	0.005	0.0056	0.0266	0.025	0.0247	0.0247	0.0238	0.0228	0.0225	0.0225	0.0222	0.0238	0.0216
0.0009	0.0013	0.0016	0.0016	0.0009	0.0006	0.0013	0.0009	0.0016	0.0013	0.0019	0.0103	0.0078	0.0075	0.0087	0.0078	0.0047	0.0053	0.0047	0.0044	0.005	0.0038
0.0038	0.0034	0.0031	0.0022	0.0025	0.0031	0.0025	0.0034	0.0031	0.0031	0.0031	0.0359	0.0353	0.0287	0.0263	0.0244	0.0219	0.0216	0.0209	0.0216	0.0194	0.0191
0.0053	0.0031	0.0028	0.0019	0.0038	0.0034	0.0034	0.0028	0.0028	0.0034	0.0034	0.0356	0.0344	0.0291	0.0262	0.0238	0.0216	0.0219	0.02	0.0219	0.0194	0.0194
0.4506	0.365	0.3259	0.3138	0.2978	0.2925	0.2834	0.2784	0.2813	0.2784	0.28	6.8453	4.925	4.0781	3.9347	4.0022	3.9722	3.9247	3.9309	3.915	3.9172	3.7778
0.0259	0.0203	0.0194	0.0181	0.0188	0.0206	0.0197	0.0194	0.0203	0.0194	0.0206	0.3869	0.3166	0.2834	0.2722	0.265	0.2578	0.2547	0.2462	0.2472	0.2528	0.2366
0.1291	0.1203	0.1066	0.1044	0.0956	0.0903	0.0906	0.0853	0.0856	0.0894	0.0872	1.8403	1.5931	1.3972	1.2806	1.2491	1.2181	1.1959	1.2003	1.1878	1.1566	1.1619
0.05	0.0422	0.0384	0.0362	0.0353	0.0334	0.0328	0.0328	0.0322	0.0331	0.0325	0.6559	0.4975	0.4375	0.4081	0.3881	0.3678	0.3541	0.3488	0.3444	0.3378	0.3322
0.0806	0.0672	0.0622	0.0588	0.0587	0.0547	0.0556	0.0541	0.0541	0.0534	0.0525	1.2022	0.9475	0.8553	0.8241	0.7809	0.7569	0.7275	0.7269	0.7231	0.7	0.6928
0.0478	0.0419	0.0388	0.0347	0.0328	0.0309	0.0306	0.03	0.0306	0.0319	0.0297	0.7031	0.5675	0.4934	0.4691	0.4263	0.3978	0.3841	0.3869	0.3816	0.37	0.3675
0.0863	0.0725	0.0659	0.0625	0.06	0.0581	0.06	0.0578	0.0578	0.0563	0.0575	1.2841	0.9731	0.86	0.7944	0.7716	0.7397	0.7184	0.7134	0.7066	0.6863	0.6816
0.0406	0.0409	0.0372	0.0331	0.0334	0.0309	0.0306	0.0313	0.0291	0.0297	0.0316	0.4691	0.3516	0.3391	0.2684	0.2647	0.2519	0.2494	0.2447	0.2447	0.2409	0.235
0.0109	0.0109	0.0109	0.0097	0.0091	0.0084	0.0084	0.0094	0.0103	0.0088	0.0106	0.2228	0.1941	0.1725	0.1603	0.1531	0.1453	0.1438	0.1453	0.1447	0.1403	0.1384
0.0188	0.0181	0.0188	0.0172	0.0169	0.015	0.0172	0.0184	0.0175	0.0175	0.0172	0.0747	0.0769	0.0728	0.0713	0.0709	0.0697	0.0681	0.0678	0.0691	0.0675	0.0662
0.0634	0.0537	0.0513	0.0519	0.05	0.0478	0.0497	0.0516	0.0491	0.0503	0.05	0.7994	0.6587	0.59	0.5722	0.5462	0.5516	0.5406	0.5425	0.5403	0.5225	0.525
0.055	0.0531	0.0522	0.0469	0.0463	0.0466	0.0462	0.0447	0.0444	0.0428	0.0459	0.4375	0.3606	0.3381	0.3297	0.3216	0.3284	0.3253	0.3206	0.3172	0.3091	0.3078
0.0691	0.0641	0.0622	0.0612	0.0603	0.0559	0.0561	0.0575	0.0578	0.0572	0.0569	0.3328	0.2806	0.2622	0.2653	0.2534	0.26	0.26	0.2591	0.2594	0.2497	0.2506
0.0431	0.0394	0.0406	0.0359	0.0372	0.0344	0.035	0.0356	0.035	0.0347	0.0359	0.2475	0.1991	0.1844	0.1828	0.1747	0.1797	0.1797	0.1759	0.1766	0.1691	0.1691
0.0681	0.0656	0.0669	0.0631	0.0647	0.0569	0.0575	0.0581	0.0578	0.0603	0.0575	0.3137	0.2653	0.2506	0.2472	0.2419	0.2431	0.2466	0.2422	0.2394	0.2344	0.235
0.0522	0.0464	0.0438	0.0415	0.0404	0.0392	0.0387	0.0384	0.0383	0.0384	0.0388	0.5437	0.4308	0.3794	0.3615	0.3556	0.3481	0.3441	0.3428	0.3410	0.3362	0.3295
0.0522	0.0464	0.0438	0.0415	0.0404	0.0392	0.0388	0.0384	0.0383	0.0384	0.0388	0.5437	0.4308	0.3794	0.3615	0.3556	0.3481	0.3441	0.3429	0.341	0.3362	0.3295

PCT

50% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balanceadas y 99% coverage											100 muestras modo RSM										
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
0.2506	0.2328	0.23	0.2206	0.2178	0.2106	0.215	0.2172	0.2159	0.215	0.2178	0.7334	0.7334	0.7275	0.7175	0.7113	0.7138	0.7009	0.6956	0.7063	0.7141	0.7053
0.0241	0.0238	0.0225	0.0244	0.0247	0.0247	0.0234	0.0234	0.0247	0.0234	0.0241	0.1691	0.1478	0.1431	0.1359	0.1331	0.1297	0.1319	0.1303	0.13	0.1278	0.1278
0.0019	0.0016	0.0016	0.0022	0.0022	0.0028	0.0025	0.0031	0.0034	0.0041	0.0038	0.0253	0.0241	0.0228	0.0237	0.0241	0.015	0.0147	0.015	0.0147	0.0134	0.0141
0.0047	0.0053	0.0047	0.0056	0.0053	0.0063	0.0056	0.0062	0.0075	0.0075	0.0059	0.0588	0.0481	0.0441	0.0387	0.0378	0.0375	0.0378	0.0381	0.0362	0.0356	0.0347
0.0075	0.0059	0.0066	0.0053	0.0069	0.0066	0.0075	0.0078	0.0091	0.0087	0.0081	0.0572	0.0469	0.0416	0.0406	0.0375	0.0359	0.0356	0.0353	0.035	0.0337	0.0337
0.0075	0.0078	0.0066	0.0069	0.0081	0.0081	0.0072	0.0081	0.0088	0.0094	0.01	0.0575	0.0494	0.0444	0.0437	0.0384	0.0397	0.0378	0.0387	0.0375	0.0394	0.0384
0.0062	0.0062	0.0072	0.0069	0.0062	0.0053	0.0059	0.0078	0.0078	0.0088	0.0075	0.0488	0.0462	0.0406	0.0344	0.0341	0.0331	0.0328	0.0337	0.0331	0.0328	0.0322
0.0044	0.0047	0.0038	0.0044	0.0034	0.0047	0.0053	0.0041	0.0063	0.0047	0.0056	0.0566	0.0472	0.0416	0.0409	0.0359	0.0338	0.0328	0.0322	0.0316	0.0325	0.03
0.0034	0.0034	0.0038	0.0034	0.0034	0.0044	0.0053	0.0053	0.0053	0.0069	0.0078	0.0809	0.0669	0.06	0.0534	0.0534	0.0478	0.0469	0.0444	0.0437	0.0428	0.0441
0.0053	0.0047	0.0044	0.0047	0.0044	0.0056	0.005	0.0047	0.0066	0.0062	0.0066	0.0894	0.0763	0.065	0.0622	0.06	0.0556	0.0541	0.0513	0.0516	0.0503	0.0491
0.0075	0.0078	0.0062	0.0063	0.0069	0.0091	0.0084	0.0069	0.0091	0.0094	0.0087	0.0522	0.0447	0.0388	0.0363	0.0356	0.0325	0.0325	0.0328	0.0309	0.0316	0.0306

75% Atributos																					
0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %	0,00 %	10,00 %	20,00 %	30,00 %	40,00 %	50,00 %	60,00 %	70,00 %	80,00 %	90,00 %	100,00 %
Balancedas y 99% coverage											100 muestras modo RSM										
78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0.3066	0.28	0.2734	0.2603	0.2569	0.25	0.2487	0.2462	0.2447	0.2425	0.2428	1.0213	1.0106	1.0147	1.0156	1.0047	1.0131	1.0259	1.0162	1.0097	1.02	1.0094
0.03	0.0306	0.0306	0.0297	0.0291	0.0288	0.0284	0.0266	0.0287	0.0287	0.0294	0.2363	0.2125	0.1991	0.1909	0.1903	0.1881	0.1881	0.1841	0.1844	0.1878	0.1838
0.0019	0.0019	0.0016	0.0022	0.0016	0.0019	0.0031	0.0019	0.0022	0.0034	0.0022	0.0281	0.0259	0.0269	0.0247	0.0262	0.0181	0.0169	0.0178	0.0175	0.0181	0.0188
0.0056	0.005	0.0053	0.0044	0.0056	0.0047	0.0056	0.0059	0.0062	0.0066	0.0075	0.0666	0.055	0.0513	0.0481	0.0459	0.0456	0.045	0.0447	0.0428	0.0416	0.0422
0.0072	0.0063	0.0066	0.0078	0.0075	0.0078	0.0069	0.0066	0.0072	0.0087	0.0084	0.0653	0.0541	0.0478	0.0481	0.0437	0.0428	0.0416	0.0416	0.0406	0.0391	0.04
0.0081	0.0081	0.0084	0.0072	0.0072	0.0081	0.0081	0.0075	0.0084	0.0075	0.0088	0.0719	0.0616	0.0522	0.0506	0.0506	0.0478	0.0506	0.0481	0.0475	0.0466	0.0456
0.0069	0.0063	0.0069	0.0063	0.0066	0.005	0.0069	0.0069	0.0075	0.0084	0.0084	0.0562	0.0516	0.0469	0.0409	0.0403	0.0394	0.0394	0.04	0.0391	0.0375	0.0388
0.0056	0.005	0.005	0.0047	0.005	0.0047	0.005	0.0053	0.0062	0.0059	0.0063	0.0666	0.0575	0.0531	0.0481	0.0462	0.0453	0.0431	0.0441	0.0419	0.0437	0.0422
0.0066	0.0056	0.0059	0.0056	0.0063	0.0047	0.0062	0.0047	0.0063	0.0069	0.0062	0.1016	0.0869	0.0803	0.0722	0.0672	0.0634	0.0609	0.0597	0.0553	0.0566	0.0525
0.0075	0.0059	0.0056	0.0062	0.0053	0.0063	0.0056	0.0066	0.0053	0.0062	0.0081	0.1106	0.0981	0.0894	0.0803	0.075	0.0728	0.0694	0.0684	0.0659	0.0653	0.0628
0.0091	0.0094	0.01	0.0091	0.0091	0.0091	0.0094	0.0084	0.01	0.01	0.0097	0.0694	0.0581	0.0547	0.0509	0.0441	0.0437	0.0428	0.0425	0.0416	0.0394	0.0394
0.0072	0.0075	0.0072	0.0066	0.0044	0.0069	0.0066	0.0066	0.0078	0.0069	0.0078	0.0434	0.0409	0.03	0.0281	0.025	0.025	0.0241	0.0241	0.0234	0.0234	0.0225
0.005	0.005	0.0056	0.0056	0.0066	0.0056	0.0053	0.005	0.0081	0.0062	0.0059	0.0306	0.0303	0.0191	0.0187	0.0191	0.0159	0.0156	0.0159	0.0178	0.0156	0.015
0.005	0.0059	0.0059	0.0053	0.005	0.0044	0.0053	0.0056	0.0053	0.0059	0.005	0.0588	0.0559	0.0506	0.0444	0.0441	0.0419	0.0409	0.0419	0.0422	0.0422	0.0416
0.0047	0.0044	0.0047	0.0053	0.0044	0.0056	0.005	0.0047	0.0062	0.0066	0.0069	0.0272	0.0253	0.0266	0.025	0.025	0.0244	0.0237	0.0241	0.0247	0.0237	0.0225
0.0009	0.0016	0.0016	0.0013	0.0013	0.0016	0.0012	0.0019	0.0025	0.0025	0.0025	0.0094	0.0078	0.0084	0.0066	0.0075	0.0047	0.0044	0.0053	0.0053	0.005	0.005
0.0028	0.0025	0.0038	0.0031	0.0031	0.0034	0.0034	0.0038	0.005	0.005	0.0041	0.0353	0.0331	0.0272	0.0256	0.0228	0.0219	0.0203	0.0212	0.0209	0.0203	0.0191
0.0012	0.0025	0.0038	0.0019	0.0031	0.0028	0.0031	0.0038	0.0028	0.0053	0.0044	0.0366	0.0356	0.0294	0.0241	0.0247	0.0228	0.0216	0.0203	0.0219	0.0209	0.02
0.3872	0.3269	0.2969	0.2734	0.2644	0.2562	0.2553	0.2519	0.2525	0.2538	0.2488	6.5016	4.7222	3.9534	3.8378	3.8128	3.7662	3.7809	3.7909	3.7441	3.7075	3.7516
0.0247	0.0216	0.0194	0.0197	0.0188	0.0206	0.0206	0.02	0.0194	0.0194	0.02	0.3953	0.3216	0.2856	0.2691	0.2594	0.2534	0.2466	0.2416	0.2403	0.2384	0.2341
0.1247	0.1163	0.1053	0.1012	0.0941	0.09	0.0875	0.0837	0.0859	0.0856	0.0878	1.7712	1.5525	1.3734	1.2644	1.2169	1.1859	1.1775	1.1681	1.1609	1.1634	1.1612
0.0469	0.0394	0.0362	0.0356	0.0334	0.0328	0.0313	0.0316	0.0316	0.0334	0.0328	6.6344	4.787	4.284	3.953	3.744	3.569	3.459	3.391	3.359	3.319	3.278
0.0756	0.065	0.0612	0.0562	0.0553	0.0547	0.0534	0.0534	0.0534	0.0525	0.0541	1.1825	0.92	0.8381	0.7872	0.7534	0.7281	0.7141	0.7022	0.6944	0.6834	0.6813
0.0472	0.0397	0.0372	0.0344	0.0316	0.0309	0.0306	0.03	0.0306	0.0316	0.0297	0.6788	0.5456	0.4856	0.44	0.4094	0.3916	0.3819	0.3769	0.37	0.3662	0.3613
0.0838	0.0706	0.065	0.0609	0.0581	0.0572	0.0556	0.0566	0.0562	0.0547	0.0562	1.2372	0.9447	0.8372	0.7769	0.7425	0.7197	0.7025	0.6891	0.6803	0.6706	0.6678
0.0391	0.0412	0.0359	0.0334	0.0334	0.03	0.0334	0.0312	0.0313	0.0319	0.0312	0.4522	0.3444	0.3325	0.2675	0.2588	0.2494	0.2453	0.2428	0.2381	0.2391	0.2378
0.0116	0.0091	0.0106	0.0097	0.01	0.0088	0.01	0.0097	0.0097	0.0109	0.01	0.2128	0.1831	0.1681	0.1559	0.1469	0.1425	0.1403	0.1372	0.1378	0.1366	0.1359
0.0184	0.0181	0.0181	0.0166	0.0163	0.0156	0.0159	0.0156	0.0162	0.0153	0.0166	0.0753	0.0756	0.0712	0.0706	0.0716	0.0678	0.0675	0.0688	0.0675	0.0669	0.0684
0.0612	0.0494	0.0463	0.0469	0.0469	0.0475	0.0459	0.0466	0.045	0.0481	0.0475	0.8263	0.6563	0.5947	0.5575	0.5503	0.5347	0.5297	0.5237	0.5203	0.5191	0.5184
0.0491	0.0513	0.0478	0.045	0.0422	0.0419	0.0425	0.0419	0.0422	0.0428	0.0431	0.4131	0.3444	0.3325	0.3141	0.3109	0.3106	0.3062	0.3053	0.3022	0.2981	0.2956
0.06	0.06	0.0516	0.055	0.0534	0.0503	0.0509	0.0541	0.0516	0.0522	0.0522	0.3244	0.2684	0.2594	0.2531	0.2491	0.2503	0.2478	0.2456	0.2447	0.2453	0.2441
0.0394	0.0369	0.0388	0.0319	0.0347	0.0353	0.0341	0.0344	0.0328	0.0331	0.0328	0.2375	0.1866	0.1784	0.1719	0.1728	0.1697	0.1662	0.1675	0.1678	0.1641	0.1628
0.0572	0.0563	0.0575	0.0569	0.0566	0.0522	0.0534	0.05	0.0556	0.0569	0.0528	0.305	0.2556	0.2497	0.2441	0.2406	0.24	0.2366	0.2344	0.235	0.2319	0.2325
0.0469	0.0423	0.0400	0.0379	0.0369	0.0359	0.0359	0.0354	0.0359	0.0362	0.0361	0.5268	0.4182	0.3726	0.3530	0.3446	0.3377	0.3353	0.3331	0.3298	0.3276	0.3273
<b>0.0469</b>	<b>0.0423</b>	<b>0.04</b>	<b>0.0379</b>	<b>0.0369</b>	<b>0.0359</b>	<b>0.0359</b>	<b>0.0354</b>	<b>0.0359</b>	<b>0.0362</b>	<b>0.0361</b>	<b>0.5267</b>	<b>0.4182</b>	<b>0.3726</b>	<b>0.353</b>	<b>0.3446</b>	<b>0.3377</b>	<b>0.3353</b>	<b>0.3331</b>	<b>0.3298</b>	<b>0.3276</b>	<b>0.3273</b>