II Semana Doctoral
Formación en la Sociedad
del Conocimiento
2018

Aplicación de técnicas de minería de datos con software Weka

Fernando Martínez Abad

Profesor Ayudante Doctor

Métodos de Investigación y Diagnóstico en Educación

Universidad de Salamanca

ÍNDICE

INTRODUCCIÓN: EL SOFTWARE WEKA

Weka es un acrónimo de *Wakaito Environment for Knowledge Analysis*, es un entorno desarrollado inicialmente por la Universidad de Wakaito, y pensado específicamente para la aplicación y evaluación de técnicas de las conocidas comúnmente como de ***minería de datos***. En concreto, el software Weka, es un software de libre distribución construido en lenguaje Jaba bajo código abierto, que permite trabajar en el preprocesado, clasificación, agrupación, asociación, predicción y visualización de los datos disponibles, incorporando numerosos algoritmos de análisis que no proveen otros software comerciales convencionales como el SPSS. A pesar de que la entrada de datos no es excesivamente intuitiva, Weka posee la ventaja con respecto a otros proyectos como R en cuanto a que la interfaz gráfica de usuario está incluida en el propio software, permitiendo acceder y configurar todas las herramientas disponibles en esta interfaz.

En los próximos folios, veremos algunas de las técnicas básicas que se pueden aplicar en este paquete estadístico.

PREPARACIÓN DE DATOS PARA TRABAJAR EN WEKA

Como ya se ha señalado previamente, la preparación de los datos para comenzar a trabajar en Weka es la cuestión más compleja y tediosa, ya que se requiere la entrada de datos a través de un fichero en formato .arff, que no es más que un fichero en formato .txt editado bajo unas condiciones concretas.

Existen varias cuestiones que son importantes a la hora de generar el archivo de datos para Weka:

- En Weka, se permiten los archivos de entrada con valores perdidos, pero para introducirlos se debe sustituir el perdido por el símbolo '?. Por lo tanto es necesario editar el archivo antes de generar la base de datos definitiva en el formato correcto.
- La entrada de variables categóricas (ordinales o nominales) debe realizarse incorporando directamente las etiquetas de las categorías para cada caso, no los valores ficticios que se incorporan habitualmente en otros programas como SPSS.
- Todas las variables deben estar definidas en el archivo de entrada de datos, en función de si son categóricas, enteras o números reales.
- La entrada de variables numéricas continuas (con decimales) deben hacerse utilizando el punto como posición decimal.
- Las puntuaciones de los sujetos en cada variable deben ir, en el archivo de datos, separadas por comas, y cada sujeto en párrafo nuevo.

Así, puede ser interesante emplear la hoja de cálculo como medio intermedio para editar los datos inicialmente, dando los siguientes pasos:

1. Eliminar los casos con valores perdidos (a no ser que se desee imputar los datos, que quizás sea más pertinente emplear otro software inicialmente), o sustituir los valores perdidos por el símbolo '**?**'.
2. Sustituir (en caso de que sea necesario) los valores numéricos de las variables categóricas por las etiquetas correspondientes.
3. Convertir el archivo de hoja de cálculo a otro con formato .csv (separado por comas) editando los datos inicialmente en ese archivo.
4. Abrir archivo .csv con el bloc de notas y realizar, en este orden, las 2 siguientes acciones:
    a. Sustituir las comas de las posiciones decimales de las variables continuas (en caso de existir en el archivo de datos) por puntos (Control+e para que seleccione todo el texto, Control+r para que se abra la ventana de sustitución).
    b. Sustituir los ; del separador generado por el archivo .csv por puntos.
5. Archivo → Guardar como, archivo .txt

Una vez se disponga del archivo en extensión .txt con los datos de los sujetos de la muestra en el formato correcto, se deben describir en el propio archivo de texto las variables que forman parte de la base de datos, definiendo un listado completo de las mismas, en el orden exacto en el que se encuentran dispuestos los valores de los sujetos en estas variables.

En concreto, se pueden definir en el archivo de entrada 3 tipos de variables:

- Variables categóricas (ordinales o nominales): Se define de la manera que está mostrada a continuación

@attribute NOMBRE_VARIABLE **{**CATEGORÍA_1,CATEGORÍA_2,…, CATEGORÍA_n**}**

- Variables cuantitativas discretas: Se definen como variables enteras, de la siguiente forma

@attribute *NOMBRE_VARIABLE* integer

- Variables cuantitativas continuas: Se definen como variables que incluyen números reales
  @attribute *NOMBRE_VARIABLE* **real**

Como últimas cuestiones, existen dos líneas que deben incorporarse en todo caso en los archivos de datos de Weka:

- Línea inicial, que debe incorporar el nombre que queremos que tenga el archivo una vez que lo abramos en Weka:
  @relation NOMBRE_ARCHIVO

- Línea entre las variables y las puntuaciones de los sujetos, que sirve como línea divisoria entre la definición de las variables y los datos en sí. Simplemente debe incluir el código: @data

Cabe resaltar que, en ningún caso, weka permite que existan espacios dentro de una misma categoría, un nombre de una variable, o un nombre de un archivo, por lo que se recomienda emplear la barra baja para delimitar espacios.

Por ejemplo, se podría definir el siguiente archivo de datos, con una variable dicotómica, una politómica, dos variables enteras, una variable real, y 5 sujetos:

@relation COMPETENCIAS_ESO

@attribute *Género* **{**Hombre**,**Mujer**}**

@attribute Rama **{**Humanidades**,**Sociales**,**Ciencias**,**Salud**,**Ingeniería_arquitectura**}**

@attribute *edad* integer

@attribute *n_hermanos* integer

@attribute *Rendimiento* real

@data

Hombre,Humanidades,19,2,6.7

Hombre,Salud,25,?,9.1

Mujer,Salud,20,1,7.3

Hombre,Sociales,21,3,?

Hombre,Humanidades,20,0,5.0

EMPEZAR A UTILIZAR WEKA

Una vez que tenemos preparada nuestra base de datos, sólo es necesario sustituir la extensión original del archivo (.txt) por la extensión propia de los archivos Weka (.arff), iniciar weka y abrir el archivo desde el botón de menú 'open file'.

Una vez abierto el archivo, en la primera pestaña disponible (Preprocess) se desplegará una información descriptiva inicial sobre las variables incluidas en la base de datos. Al respecto, cabe destacar una cuestión terminológica empleada en el ámbito de la minería de datos en general, y en Weka en particular:

- Las variables de la base de datos son consideradas *atributos*.
- Los sujetos o casos son considerados *instancias*.

Así, si deseamos realizar en esta fase de preprocesado alguna modificación inicial sobre los datos antes de aplicar las técnicas deseadas (categorizar una variable cuantitativa, generar una variable de agrupación a partir de un cluster, aplicar una técnica de remuestreo, etc.), debemos tener en cuenta que Weka utiliza la nomenclatura general 'atributo' (variable) e 'instancia' (sujeto). Además, existen unas técnicas de preprocesado supervisadas (algoritmos que emplean una variable criterio o **atributo clase**) y otras no supervisadas (técnicas en las que los cambios se realizan en base a un criterio o criterios concretos establecidos por el investigador). En este caso, nos va a interesar de manera generalizada aplicar **filtros no supervisados**.

Una vez que tengamos los datos definidos y preparados como deseamos, podemos pasar a aplicar la técnica deseada (asociación, clustering o clasificación).

REGLAS DE ASOCIACIÓN

Definición

Las reglas de asociación se emplean para buscar relaciones entre los sucesos o acciones que se pueden considerar. Estas reglas integran algoritmos que tienen como objetivo identificar la ocurrencia conjunta de varios sucesos, permitiendo extraer información acerca de cómo la ocurrencia o no ocurrencia de algunos sucesos puede inducir la aparición de otros.

Estas técnicas se emplean generalmente en el estudio exploratorio de un conjunto muy numeroso de variables categóricas o discretas. El algoritmo detectará las asociaciones existentes entre las variables que tengan una tasa de cobertura más alta, es decir, aquellas reglas de asociación que se cumplan en nuestra muestra en un mayor porcentaje de los casos.

El algoritmo fundamental que incorpora Weka al respecto es el algoritmo *A priori,* que establece las reglas teniendo en cuenta el **soporte** de los datos a la regla y la **confianza** de la propia regla. El soporte se refiere al número de instancias que están incluidas en la regla, y la confianza al porcentaje de instancias, de entre el total del soporte, que cumplen la regla, es decir, el número de casos que predice la regla correctamente:

$$soporte(A \Rightarrow B) = P(A \cap B)$$

$$confianza(A \Rightarrow B) = P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$

A nivel general Weka entiende que, bajo un nivel de confianza dado, una regla será más interesante cuanto mayor sea el soporte bajo el que esté generada. Así, El algoritmo comienza buscando las reglas que alcancen una mayor confianza cuyo soporte sea superior, bajando dentro de un mismo nivel de confianza el nivel de soporte hasta el límite fijado. Cuando se alcanza el límite mínimo de soporte, se vuelven a generar normas para el siguiente nivel de confianza inferior.

Procedimiento

En Weka, para aplicar un algoritmo de asociación, debemos ir a la pestaña 'associate'. Ahí, haciendo clic en el botón 'Choose', podremos seleccionar el algoritmo deseado (en este caso utilizaremos a priori). Posteriormente, haciendo clic sobre la ventana a la derecha del botón Choose, podremos seleccionar los parámetros de la aplicación, los principales parámetros que podemos establecer son:

- upperBoundMinSupport: Porcentaje máximo de soporte sobre el total de sujetos.
- lowerBoundMinSupport: Porcentaje mínimo de soporte sobre el total de sujetos.
- Delta: Reducción del soporte en cada iteración.
- minMetric: Porcentaje mínimo de confianza para generar la regla.
- numRules: Número de reglas generadas por el procedimiento.

- metricType: estadístico empleado para la toma de decisiones sobre la generación de las reglas:
    a. Confidence: se generan las n reglas soportadas con un mayor índice de confianza.
    b. Lift: se generan las n reglas soportadas que tengan un valor mayor del índice de confianza dividido entre el número de casos cubiertos por la norma. Este índice es independiente de la tasa de soporte.
    c. Leverage: Se generan las n reglas soportadas de las n diferencias máximas entre el porcentaje de confianza y el porcentaje esperado si existiera independencia entre la regla y el soporte.
    d. Conviction: Se generan las n reglas soportadas a partir de los n valores máximos alcanzados en el cálculo de una probabilidad condicionada entre la regla y el soporte.
- significanceLevel: Si se mantiene en -1 no se utiliza. Se puede establecer un nivel de significación mínimo para que se genere la norma.
- treatZeroAsMissing: Si se señala 'True', el programa no considerará las respuestas '0' como categorías a tener en cuenta a la hora de generar una norma.

Una vez seleccionadas las cuestiones deseadas, simplemente haremos clic en 'Start', y el programa comenzará a trabajar. En la ventana de salida nos aparecerán los resultados con las normas establecidas.

ALGORITMOS DE CLUSTERING

Definición

Las técnicas de clustering se emplean para identificar tendencias comunes por parte de grupos de sujetos en las puntuaciones obtenidas en un conjunto de variabes, de manera que se puedan establecer gruposde sujetos similares entre sí y diferentes con respecto al resto. Estas técnicas sirven generalmente para segmentar a un conjunto de sujetos en grupos en función e sus características personales, buscando grupos sujetos que se comporten de manera similar entre sí, esto es, en los que la homogeneidad intragrupo y la heterogeneidad intergrupo sean máximas.

Existen gran cantidad de algoritmos de clustering, siendo los 3 más habitualmente empleados en Weka los siguientes:

1. Clustering Numérico (k-medias)

Este algoritmo se puede emplear cuando las variables empleadas para llevar a cabo la agrupación son numéricas, no siendo apropiado en otros casos. Es el algoritmo que habitualmente se emplea en SPSS para generar grupos. Su cálculo es muy simple, ya que simplemente asigna al sujeto al clúster al que esté más cercano, conforme a la distancia euclidea entre el sujeto y el centroide del clúster, calculado a partir de las puntuaciones de todos los sujetos del grupo asignado. El proceso se itera, sujeto a sujeto, hasta que todos los sujetos se mantienen en el mismo centroide.

2. Clustering Conceptual (COBWEB)

Cuando disponemos en nuestra base de datos de variables de naturaleza no cuantitativa (ordinales o cualitativas), el algoritmo k-medias no es apropiado. En este caso, se dispone de un algoritmo como el COBWEB, que emplea el concepto de proximidad o distancia entre los elementos de la población sin emplear la simple distancia euclidea. El método COBWEB entiende los clústeres como distribuciones de probabilidad sobre el espacio de las puntuaciones de los sujetos, calculando los puntos de corte en las variables en los que se maximiza la distancia entre grupos de sujetos, generando de esta manera árboles de clasificación. El árbol resultante suele denominarse jerarquía de conceptos. En la construcción del árbol, se va a incorporando cada sujeto de manera incremental, buscando el mejor lugar o nodo para cada sujeto en el árbol.

De igual manera, se pueden incorporar variables numéricas en este algoritmo, ya queCOBWEB emplea la distribución de probabilidades de la distribución normal para calcular el corte que maximice la heterogeneidad de los sujetos.

3. Clustering Probabilístico (EM)

Los dos algoritmos anteriores presentan el mismo problema de dependencia del resultado del orden en el que estén presentados los sujetos en la base de datos, y su tendencia a sobreajustar los clústeres obtenidos en las muestras de entrenamiento. El algoritmo EM permite un

acercamiento probabilístico al problema del clústering, solucionando los mencionados problemas. Ahora, en lugar de buscar sujetos parecidos entre sí de manera iterativa, lo que se intenta es buscar el grupo de clústeres más probables dado un conjunto de puntuaciones. El algoritmo se basa en calcular las probabilidades que existen de que un sujeto tenga una puntuación en la variable, si se supiera que el sujeto es miembro de ese clúster. Así, se obtienen k distribuciones de probabilidad, una por cada uno de los k clústeres. Lo que hace el algoritmo EM es *adivinar* inicialmente los parámetros de las distribuciones para, a continuación, emplear esos parámetros para llevar a cabo el cálculo de las probabilidades de que cada sujeto pertenezca a un cluster. Posteriormente, emplea esas probabilidades para re-estimar los parámetros. Y así hasta llegar al criterio de parada establecido, en base a un valor mínimo de convergencia.

Procedimiento

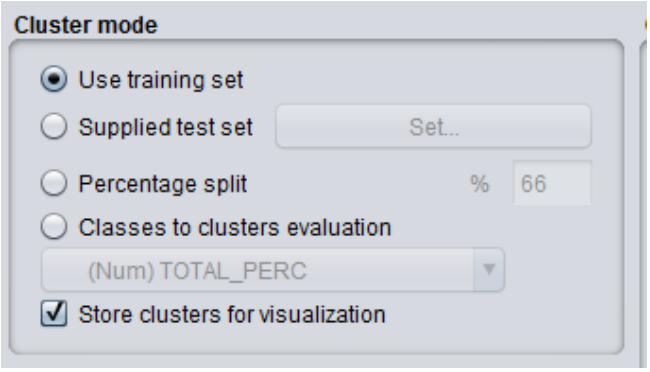
Los algoritmos de agrupación o Clustering permiten dos tipos de técnicas:

- **Exploración inicial**: testear las agrupaciones que aseguren una mayor homogeneidad intragrupo y una mayor heterogeneidad intergrupo a partir de una serie de variables de entrada.
- **Evaluación a partir de una variable criterio**: probar si un conjunto de variables de entrada puede emplearse como predictor de una variable de clase o variable criterio, generando unas agrupaciones conforme al criterio en base a las puntuaciones obtenidas en las variables predictoras.

En ambos casos, se pueden generar modelos empleando la muestra completa como muestra de entrenamiento, o estableciendo algún tipo de control del sobreajuste a partir de validaciones. La validación de los datos a partir de submuestras es altamente recomendable, ya que los procedimientos de Clustering (al igual que las técnicas de clasificación) tienden a generar modelos que sobreestiman la verdadera relación entre la variable criterio y las variables predictoras. Las principales técnicas para la validación de datos que nos ofrece Weka son:

- **División de la muestra:** Se establece de antemano una submuestra que será considerada muestra de entrenamiento (muestra a partir de la que se genera el modelo con las normas de agrupación), a partir de la que se genera el modelo principal. Ese modelo es contrastado a partir de la otra submuestra, que es considerada simplemente para esta validación.
- **Validación cruzada (sólo disponible en las técnicas de clasificación)**: Se genera en primer lugar el modelo de clasificación, contando con la muestra completa como muestra de entrenamiento. Posteriormente, se divide a la muestra en k submuestras y el modelo es testeado en todas ellas. Los resultados de la validación se muestran indicando una media aritmética de los índices de ajuste obtenidos en cada una de las k submuestras.

Los 3 algoritmos de Clustering señalados tienen un funcionamiento similar. Siempre tendremos disponible la ventana 'Cluster Mode', que permite las siguientes opciones:



- *Use training set:* Se trabajará con la muestra de entrenamiento como muestra completa.
- *Supplied test set:* Se trabajará con la muestra completa como muestra de entrenamiento y los datos se validan en otra muestra incorporada en otro archivo.
- *Percentaje Split:* Técnica de validación de división de la muestra
- *Classes to clusters evaluation*: Emplear alguna variable categórica como variable criterio a partir de la que comparar la asignación de clústeres.

Otra opción disponible siempre es *ignore attributes*, en donde podemos señalar algunas variables de la base de datos que no queremos que entren a formar parte de las variables predictoras para generar el modelo. Finalmente, se dispone del botón *Start* y el botón *Stop*, el primero sirve para comenzar a generar el modelo deseado, y el segundo para parar el cómputo de un modelo (nos sirve cuando pedimos un modelo excesivamente complejo o que no converge, y queremos parar el trabajo de la máquina de Weka). Cada prueba que realicemos nos aparecerá como un listado en *Result list*, y la información concreta de la prueba aparecerá en *Clusterer output*.

Para definir el tipo de algoritmo que queremos aplicar, haremos clic en *Choose*, seleccionaremos la técnica deseada, y posteriormente haciendo clic sobre la ventana de la derecha, en la que se despliega la información sobre el algoritmo seleccionado, podremos establecer algunos parámetros concretos. Estos parámetros son diferentes en función del algoritmo seleccionado, y se establecen unos valores por defecto en todos ellos. Los parámetros principales son los siguientes:

1. Clustering Numérico (k-medias, sólo para datos numéricos, reales o enteros)

- *InitializationMethod:* Punto de partida para iniciar el algoritmo, normalmente de manera aleatoria, tomando una semilla.
- *DisplayStdDevs:* Mostrar la desviación típica además de la media en los valores asignados en cada variable a cada cluster.
- *DistanceFunction:* Método para calcular las distancias entre los sujetos y los centroides, normalmente Distancia euclidiana.
- *fastDistanceCalc:* Para reducir el tiempo de cómputo, se calculan las distancias a partir de puntos de corte en lugar de las sumas de cuadrados de las distancias.
- *maxIterations:* Máximo de iteraciones permitidas en las que el modelo debe converger
- *numClusters:* Número de clústeres que serán establecidos en el modelo.

- *numExecutionSlots:* Número de ranuras de expansión empleadas para el cómputo (depende de cada ordenador, si no se sabe, mantener 1).

2. Clustering Conceptual (COBWEB)

- *Acuity:* Desviación típica mínima para los atributos numéricos.
- *Cutoff:* Umbral a partir del que se establece un corte o no (a mayor cutoff, mayor es el umbral, por lo que se establecen menos cortes en el árbol).

3. Clustering Probabilístico (EM)

- *maxIterations:* Máximo de iteraciones permitidas en las que el modelo debe converger.
- *maximumNumberOfClusters:* El algoritmo EM puede realizar un testeo del número de clústeres óptimo para los datos, si se selecciona esta opción, se puede establecer el número de clústeres máximo tolerable en el modelo.
- *nunClusters:* Se puede establecer aquí un número concreto de Clústeres a extraer.
- *numExecutionSlots:* Número de ranuras de expansión empleadas para el cómputo (depende de cada ordenador, si no se sabe, mantener 1).
- *numFlods:* Número de submuestras empleado en la prueba del número óptimo de clústeres.
- *numKMeansRuns:* Veces que se itera el algoritmo k-medias a la hora de generar el modelo.

Una vez se ha obtenido el modelo deseado, se puede guardar una nueva variable en la base de datos con la asignación obtenida haciendo clic con el botón derecho del ratón sobre el modelo en *Results lists*, seleccionar *Visualice clusters assignments*, y en la ventana emergente hacer clic en *Save*. El archivo guardado incluirá las variables de la base de datos original y una nueva variable con la asignación llevada a cabo por la técnica de clústering aplicada.

ALGORITMOS DE CLASIFICACIÓN

Definición

Los métodos de clasificación son las técnicas de minería de datos empleados más frecuentemente en la investigación en Ciencias Sociales y de la Educación. Muchas veces, estos algoritmos son aplicados tras haber realizado una exploración previa con los datos, a partir de clustering o reglas de asociación, como una manera de refinar y aportar una mayor especificidad a la información obtenida en las fases previas. El objetivo de estas técnicas es construir un modelo predictivo, que sea capaz de establecer con la mayor precisión posible en qué valor se encontrará el sujeto en la variable criterio a partir de la información obtenida con otras variables que podrían ser consideradas como predictoras. Entre los algoritmos más extendidos se pueden destacar los siguientes:

1. Clasificador 'OneR'

Este es uno de los clasificadores más sencillos y rápidos. Sin embargo, los resultados que devuelve resultan ser muy cercanos a los obtenidos a partir de otros algoritmos mucho más complejos. Este algoritmo selecciona la variable predictora que mejor "explica" la variable criterio seleccionada. Si hay variables numéricas, busca los umbrales para generar la regla con mejor tasa de aciertos.

2. Clasificador J48: Árboles de decisión

El algoritmo J48, integrado en Weka, es uno de los algoritmos de minería de datos más extendido en los estudios que incluyen algoritmos de clasificación. Entre los parámetros estimados bajo este procedimiento, destaca nivel de confianza establecido para la poda del árbol generado, **confidence level**, puesto que influye notoriamente en el tamaño y capacidad de predicción del árbol construido.

Se podría explicar el clasificador de la siguiente manera: para tomar la decisión sobre el corte realizado en la iteración 'n', se busca la variable predictora y el punto de corte exacto en el que el error cometido es más bajo (tomando como criterio una variable preestablecida), siempre y cuando nos encontremos en niveles de confianza superiores a los establecidos previamente. Una vez realizado el corte, el algoritmo vuelve a repetirse, hasta que ninguna de las variables predictoras alcance un nivel de confianza superior al establecido. Se destaca la importancia de trabajar con el nivel de confianza, ya que, en caso de tener un gran número de sujetos y variables, este árbol puede resultar demasiado grande. Otra forma de limitar el tamaño del árbol es especificando el mínimo número de instancias por nodo.

3. Clasificador bayesiano NaiveBayes

En base al teorema de Bayes:

Sean A y B dos sucesos aleatorios cuyas probabilidades se denotan por p(A) y p(B) respectivamente, verificándose que p(B)>0. Supongamos conocidas las probabilidades a priori

de los sucesos A y B, es decir, p(a) y p(B), así como la probabilidad condicionada del suceso B dado el suceso A, es decir p(B|A). La probabilidad a posteriori del suceso A conocido que se verifica el suceso B, es decir p(A|B), puede calcularse a partir de la fórmula:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{k=1}^{n} P(B|A_k)P(A_k)}$$

Así, podemos aplicar las propiedades de este teorema a la clasificación en la minería de datos. En este sentido, el Bayes Naive es un clasificador bayesiano, bajo el supuesto de que las variables predictoras incluidas en el modelo son independientes en cada una de las clases.

Procedimiento

Los algoritmos de clasificación tienen diferentes características y parámetros que definir en función de los señalados anteriormente:

1. Clasificador 'OneR'

- *batchSize*: Si se realiza la agrupación de los sujetos por lotes, tamaño de cada lote.
- *minBucketSize*: Mínimo de categorías en las que se discretizan las variables numéricas incluidas en el análisis.
- *numDecimalPlaces*: Número de decimales presentados en los datos mostrados en el output.

2. Clasificador J48

- *binarySplits*: Realizar solamente cortes en dos secciones en el árbol.
- *collapseTree*: Permitir eliminar una rama que aumenta el error al generar la rama posterior.
- *ConfidenceFactor*: El nivel a partir del que se establece un corte en el árbol. A mayor factor de confianza, más cortes se establecen en el árbol.
- *minNumObj*: Número mínimo de sujetos por grupo para poder establecer un corte.
- *numDecimalPlaces*: Número de decimales presentados en los datos mostrados en el output.
- *NumFolds*: Número de submuestras en los que se divide la muestra para reducir el error, siempre y cuando se active la opción *reducedErrorPruning*.

En cuanto a los resultados que se nos muestran en este procedimiento, existen varios indicadores de la bondad de ajuste del modelo que pueden ser interesantes:

- Porcentaje de instancias bien clasificadas a nivel global.
- **Índice Kappa de Cohen**: Índice de bondad de ajuste del modelo completo. Se entiende que el modelo es apropiado si este valor supera el valor 0.7.
- **Error absoluto medio**: Estadístico que indica el error de estimación cometido. Valores más bajos indican que el modelo es más apropiado.

- **Porcentaje del área bajo la curva ROC**: La curva ROC indica, en un eje de abscisas y ordenadas, la relación entre la sensibilidad (verdaderos positivos entre el total de positivos) y especificidad (verdaderos negativos entre el total de negativos) del modelo de clasificación establecido. Se establece una curva para cada categoría, en relación con el resto de categorías, indicando la capacidad del modelo para detectar casos pertenecientes a esa categoría de la variable criterio. Este porcentaje nos indica, por lo tanto, la precisión que tiene el modelo para identificar correctamente a los sujetos de un grupo, teniendo en cuenta tanto el porcentaje de acierto en la detección de esa categoría y el porcentaje de desaciertos al identificar a sujetos de esa categoría
- **Matriz de confusión**: Tabla de contingencia que relaciona la clasificación dada por el modelo con el valor real que alcanza el sujeto en la variable criterio.
- **Precisión**: Porcentaje de instancias bien clasificadas de entre todas las seleccionadas como positivas.