



ANÁLISIS DE DATOS

Jesús García Herrero

Capítulo 1. Técnicas de Análisis de Datos en WEKA

CAPÍTULO 1. TÉCNICAS DE ANÁLISIS DE DATOS EN WEKA	1
1.1. Introducción	2
1.2. Preparación de los datos	3
1.2.1. Muestra de datos	3
1.2.2. Objetivos del análisis	4
1.3. Ejecución de WEKA	5
1.4. Preprocesado de los datos	7
1.4.1. Características de los atributos	8
1.4.2. Trabajo con Filtros. Preparación de ficheros de muestra	9
1.4.2.1. Filtros de atributos	10
1.4.2.2. Filtros de instancias	14
1.5. Visualización	15
1.5.1. Representación 2D de los datos	15
1.5.2. Filtrado “gráfico” de los datos	19
1.6. Asociación	20
1.7. Agrupamiento	25
1.7.1. Agrupamiento numérico	26
1.7.2. Agrupamiento simbólico	31
1.8. Clasificación	33
1.8.1. Modos de evaluación del clasificador	34
1.8.2. Selección y configuración de clasificadores	37
1.8.3. Predicción numérica	45
1.8.4. Aprendizaje del modelo y aplicación a nuevos datos.	51
1.9. Selección de atributos	53

1.1. Introducción

En este capítulo se presenta de forma concisa y práctica la herramienta de minería de datos WEKA. WEKA, acrónimo de *Waikato Environment for Knowledge Analysis*, es un entorno para experimentación de análisis de datos que permite aplicar, analizar y evaluar las técnicas más relevantes de análisis de datos, principalmente las provenientes del aprendizaje automático, sobre cualquier conjunto de datos del usuario. Para ello únicamente se requiere que los datos a analizar se almacenen con un cierto formato, conocido como *ARFF* (*Attribute-Relation File Format*).

WEKA se distribuye como software de libre distribución desarrollado en Java. Está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados. Estos paquetes pueden ser integrados en cualquier proyecto de análisis de datos, e incluso pueden extenderse con contribuciones de los usuarios que desarrollen nuevos algoritmos. Con objeto de facilitar su uso por un mayor número de usuarios, WEKA además incluye una interfaz gráfica de usuario para acceder y configurar las diferentes herramientas integradas.

Este capítulo tiene un enfoque práctico y funcional, pretendiendo servir de guía de utilización de esta herramienta desde su interfaz gráfica, como material complementario a la escasa documentación disponible. Para ello se obviarán los detalles técnicos y específicos de los diferentes algoritmos, que se presentan en un capítulo aparte, y se centrará en su aplicación, configuración y análisis dentro de la herramienta. Por tanto, se remite al lector al capítulo con los detalles de los algoritmos para conocer sus características, parámetros de configuración, etc. Aquí se han seleccionado algunas de las técnicas disponibles para aplicarlas a ejemplos concretos, siguiendo el acceso desde la herramienta al resto de técnicas implementadas, una mecánica totalmente análoga a la presentada a modo ilustrativo.

Para reforzar el carácter práctico de este capítulo, además se adoptará un formato de tipo tutorial, con un conjunto de datos disponibles sobre el que se irán aplicando las diferentes facilidades de WEKA. Se sugiere que el lector aplique los pasos indicados y realice los análisis sugeridos para cada técnica con objeto de familiarizarse y mejorar su comprensión. Los ejemplos seleccionados son contienen datos provenientes del campo de la enseñanza, correspondientes a alumnos que realizaron las pruebas de selectividad en los años 1993-2003 procedentes de diferentes centros de enseñanza secundaria de la comunidad de Madrid. Por tanto, esta guía ilustra la aplicación y análisis de técnicas de extracción de conocimiento sobre datos del campo de la enseñanza, aunque sería directa su traslación a cualquier otra disciplina.

1.2. Preparación de los datos

Los datos de entrada a la herramienta, sobre los que operarán las técnicas implementadas, deben estar codificados en un formato específico, denominado *Attribute-Relation File Format* (extensión ".arff"). La herramienta permite cargar los datos en tres soportes: fichero de texto, acceso a una base de datos y acceso a través de internet sobre una dirección URL de un servidor web. En nuestro caso trabajaremos con ficheros de texto. Los datos deben estar dispuestos en el fichero de la forma siguiente: cada instancia en una fila, y con los atributos separados por comas. El formato de un fichero arff sigue la estructura siguiente:

```
% comentarios
@relation NOMBRE_RELACION
@attribute r1 real
@attribute r2 real ...
...
@attribute i1 integer
@attribute i2 integer
...
@attribute s1 {v1_s1, v2_s1,...vn_s1}
@attribute s2 {v1_s1, v2_s1,...vn_s1}
...
@data
DATOS
```

por tanto, los atributos pueden ser principalmente de dos tipos: numéricos de tipo real o entero (indicado con las palabra *real* o *integer* tras el nombre del atributo), y simbólicos, en cuyo caso se especifican los valores posibles que puede tomar entre llaves.

1.2.1. Muestra de datos

El fichero de datos objeto de análisis en esta guía contiene muestras correspondientes a 18802 alumnos presentados a las pruebas de selectividad y los resultados obtenidos en las pruebas. Los datos que describen cada alumno contienen la siguiente información: año, convocatoria, localidad del centro, opción cursada (de 5 posibles), calificaciones parciales obtenidas en lengua, historia, idioma y las tres asignaturas opcionales, así como la designación de las asignaturas de idioma y las 3 opcionales cursadas, calificación en el bachillerato, calificación final y si el alumno se presentó o no a la prueba. Por tanto, puede comprobarse que la cabecera del fichero de datos, "selectividad.arff", sigue el formato mencionado anteriormente:

```
@relation selectividad

@attribute Año_académico real
@attribute convocatoria {J, S}
@attribute localidad {ALPEDRETE, ARANJUEZ, ... }
```

```

@attribute opcion1a {1,2,3,4,5}
@attribute nota_Lengua real
@attribute nota_Historia real
@attribute nota_Idioma real
@attribute des_Idioma {INGLES, FRANCES, ALEMAN}
@attribute des_asig1 {BIOLOGIA, DIB.ARTISTICO_II,... }
@attribute calif_asig1 real
@attribute des_asig2 {BIOLOGIA, C.TIERRA, ...}
@attribute calif_asig2 real
@attribute des_asig3 {BIOLOGIA, C.TIERRA, ...}
@attribute calif_asig3 real
@attribute cal_prueba real
@attribute nota_bachi real
@attribute cal_final real
@attribute Presentado {SI, NO}
@data
...

```

1.2.2. Objetivos del análisis

Antes de comenzar con la aplicación de las técnicas de WEKA a los datos de este dominio, es muy conveniente hacer una consideración acerca de los objetivos perseguidos en el análisis. Como se mencionó en la introducción, un paso previo a la búsqueda de relaciones y modelos subyacentes en los datos ha de ser la comprensión del dominio de aplicación y establecer una idea clara acerca de los objetivos del usuario final. De esta manera, el proceso de análisis de datos (proceso *KDD*), permitirá dirigir la búsqueda y hacer refinamientos, con una interpretación adecuada de los resultados generados. Los objetivos, utilidad, aplicaciones, etc., del análisis efectuado no "emergen" de los datos, sino que deben ser considerados con detenimiento como primer paso del estudio.

En nuestro caso, uno de los objetivos perseguidos podría ser el intentar relacionar los resultados obtenidos en las pruebas con características o perfiles de los alumnos, si bien la descripción disponible no es muy rica y habrá que atenerse a lo que está disponible. Algunas de las preguntas que podemos plantearnos a responder como objetivos del análisis podrían ser las siguientes:

- ¿Qué características comunes tienen los alumnos que superan la prueba? ¿y los alumnos mejor preparados que la superan sin perjudicar su expediente?
- ¿existen grupos de alumnos, no conocidos de antemano, con características similares?
- ¿hay diferencias significativas en los resultados obtenidos según las opciones, localidades, años, etc.?,
- ¿la opción seleccionada y el resultado está influida depende del entorno?

- ¿se puede predecir la calificación del alumno con alguna variable conocida?
- ¿qué relaciones entre variables son las más significativas?

Como veremos, muchas veces el resultado alcanzado puede ser encontrar relaciones triviales o conocidas previamente, o puede ocurrir que el hecho de no encontrar relaciones significativas, lo puede ser muy relevante. Por ejemplo, saber después de un análisis exhaustivo que la opción o localidad no condiciona significativamente la calificación, o que la prueba es homogénea a lo largo de los años, puede ser una conclusión valiosa, y en este caso "tranquilizadora".

Por otra parte, este análisis tiene un enfoque introductorio e ilustrativo para acercarse a las técnicas disponibles y su manipulación desde la herramienta, dejando abierto para el investigador llevar el estudio de este dominio a resultados y conclusiones más elaboradas.

1.3. Ejecución de WEKA

WEKA se distribuye como un fichero ejecutable comprimido de java (fichero "jar"), que se invoca directamente sobre la máquina virtual JVM. En las primeras versiones de WEKA se requería la máquina virtual Java 1.2 para invocar a la interfaz gráfica, desarrollada con el paquete gráfico de Java *Swing*. En el caso de la última versión, WEKA 3-4, que es la que se ha utilizado para confeccionar estas notas, se requiere Java 1.3 o superior. La herramienta se invoca desde el intérprete de Java, en el caso de utilizar un entorno windows, bastaría una ventana de comandos para invocar al intérprete Java:



```

C:\>java -jar weka.jar_

```

Una vez invocada, aparece la ventana de entrada a la interfaz gráfica (*GUI-Chooser*), que nos ofrece cuatro opciones posibles de trabajo:

- | | |
|--|--|
| | <ul style="list-style-type: none"> • Simple CLI: la interfaz "Command-Line Interfaz" es simplemente una ventana de comandos java para ejecutar las clases de WEKA. La primera distribución de WEKA no disponía de interfaz gráfica y las clases de sus |
|--|--|

	<p>paquetes se podían ejecutar desde la línea de comandos pasando los argumentos adecuados.</p> <ul style="list-style-type: none"> • Explorer: es la opción que permite llevar a cabo la ejecución de los algoritmos de análisis implementados sobre los ficheros de entrada, una ejecución independiente por cada prueba. Esta es la opción sobre la que se centra la totalidad de esta guía. • Experimenter: esta opción permite definir experimentos más complejos, con objeto de ejecutar uno o varios algoritmos sobre uno o varios conjuntos de datos de entrada, y comparar estadísticamente los resultados • KnowledgeFlow: esta opción es una novedad de WEKA 3-4 que permite llevar a cabo las mismas acciones del "Explorer", con una configuración totalmente gráfica, inspirada en herramientas de tipo "data-flow" para seleccionar componentes y conectarlos en un proyecto de minería de datos, desde que se cargan los datos, se aplican algoritmos de tratamiento y análisis, hasta el tipo de evaluación deseada.
--	--

En esta guía nos centraremos únicamente en la segunda opción, *Explorer*. Una vez seleccionada, se crea una ventana con 6 pestañas en la parte superior que se corresponden con diferentes tipos de operaciones, en etapas independientes, que se pueden realizar sobre los datos:

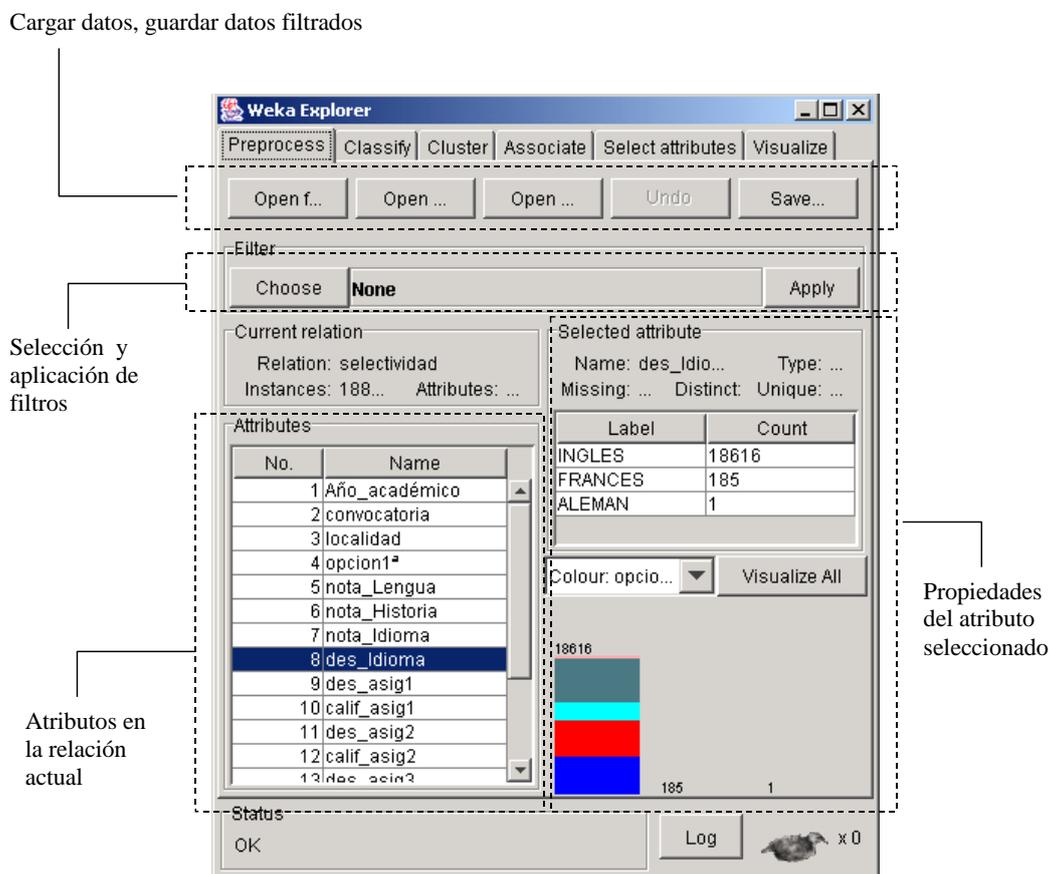
- **Preprocess:** selección de la fuente de datos y preparación (filtrado).
- **Classify:** Facilidades para aplicar esquemas de clasificación, entrenar modelos y evaluar su precisión
- **Cluster:** Algoritmos de agrupamiento
- **Associate:** Algoritmos de búsqueda de reglas de asociación
- **Select Attributes:** Búsqueda supervisada de subconjuntos de atributos representativos
- **Visualize:** Herramienta interactiva de presentación gráfica en 2D.

Además de estas pestañas de selección, en la parte inferior de la ventana aparecen dos elementos comunes. Uno es el botón de "**Log**", que al activarlo

presenta una ventana textual donde se indica la secuencia de todas las operaciones que se han llevado a cabo dentro del “Explorer”, sus tiempos de inicio y fin, así como los mensajes de error más frecuentes. Junto al botón de log aparece un icono de actividad (el pájaro WEKA, que se mueve cuando se está realizando alguna tarea) y un indicador de status, que indica qué tarea se está realizando en este momento dentro del Explorer.

1.4. Preprocesado de los datos

Esta es la parte primera por la que se debe pasar antes de realizar ninguna otra operación, ya que se precisan datos para poder llevar a cabo cualquier análisis. La disposición de la parte de preprocesado del *Explorer*, **Preprocess**, es la que se indica en la figura siguiente.



Como se indicó anteriormente, hay tres posibilidades para obtener los datos: un fichero de texto, una dirección URL o una base de datos, dadas por las opciones: **Open file**, **Open URL** y **Open DB**. En nuestro caso utilizaremos siempre los datos almacenados en un fichero, que es lo más rápido y cómodo de utilizar. La preparación del fichero de datos en formato ARFF ya se describió en la sección 1.2.

En el ejemplo que nos ocupa, abra el fichero “selectividad.arff” con la opción **Open File**.

1.4.1. Características de los atributos

Una vez cargados los datos, aparece un cuadro resumen, *Current relation*, con el nombre de la relación que se indica en el fichero (en la línea @relation del fichero arff), el número de instancias y el número de atributos. Más abajo, aparecen listados todos los atributos disponibles, con los nombres especificados en el fichero, de modo que se pueden seleccionar para ver sus detalles y propiedades.

The screenshot shows the Weka Explorer interface. The 'Current relation' section displays: Relation: selectividad, Instances: 18802, Attributes: 18. The 'Selected attribute' section shows: Name: Año académico, Type: Nume..., Missing: 0 (0%), Distinct: ..., Unique: 0 (0%). A table of statistics for the selected attribute is shown below:

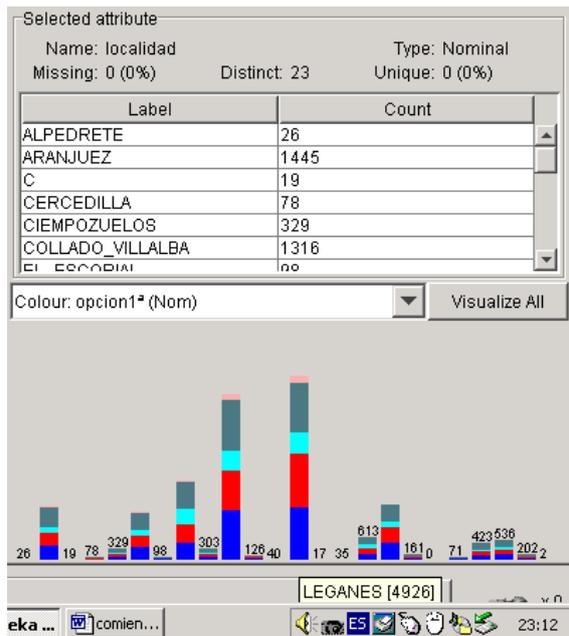
Statistic	Value
Minimum	1993
Maximum	2002
Mean	1999.417
StdDev	2.182

Below the statistics is a histogram for the attribute 'Año académico' (Nom). The x-axis ranges from 1993 to 2002. The histogram shows the distribution of values for this attribute. A 'Visualize All' button is visible at the bottom right of the histogram area.

En la parte derecha aparecen las propiedades del atributo seleccionado. Si es un atributo simbólico, se presenta la distribución de valores de ese atributo (número de instancias que tienen cada uno de los valores). Si es numérico aparece los valores máximo, mínimo, valor medio y desviación estándar. Otras características que se destacan del atributo seleccionado son el tipo (*Type*), número de valores distintos (*Distinct*), número y porcentaje de instancias con valor desconocido para el atributo (*Missing*, codificado en el fichero arff con “?”), y valores de atributo que solamente se dan en una instancia (*Unique*).

Además, en la parte inferior se presenta gráficamente el histograma con los valores que toma el atributo. Si es simbólico, la distribución de frecuencia de los valores, si es numérico, un histograma con intervalos uniformes. En el histograma se puede presentar además con colores distintos la distribución de un segundo atributo para cada valor del atributo visualizado. Por último, hay un botón que permite visualizar los histogramas de todos los atributos simultáneamente.

A modo de ejemplo, a continuación mostramos el histograma por localidades, indicando con colores la distribuciones por opciones elegidas.



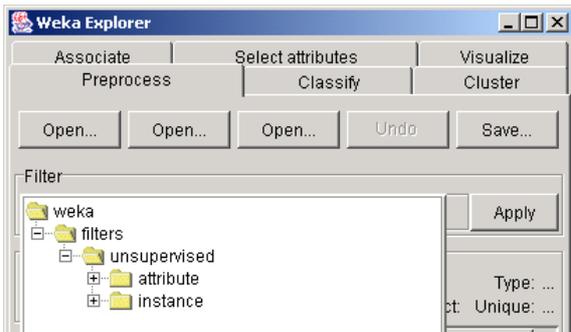
Se ha seleccionado la columna de la localidad de Leganés, la que tiene más instancias, y donde puede verse que la proporción de las opciones científicas (1 y 2) es superior a otras localidades, como Getafe, la segunda localidad en número de alumnos presentados.

Visualice a continuación los histogramas de las calificaciones de bachillerato y calificación final de la prueba, indicando como segundo atributo la convocatoria en la que se presentan los alumnos.

1.4.2. Trabajo con Filtros. Preparación de ficheros de muestra

WEKA tiene integrados filtros que permiten realizar manipulaciones sobre los datos en dos niveles: atributos e instancias. Las operaciones de filtrado pueden aplicarse “en cascada”, de manera que cada filtro toma como entrada el conjunto de datos resultante de haber aplicado un filtro anterior. Una vez que se ha aplicado un filtro, la relación cambia ya para el resto de operaciones llevadas a cabo en el *Experimenter*, existiendo siempre la opción de deshacer la última operación de filtrado aplicada con el botón **Undo**. Además, pueden guardarse los resultados de aplicar filtros en nuevos ficheros, que también serán de tipo ARFF, para manipulaciones posteriores.

Para aplicar un filtro a los datos, se selecciona con el botón **Choose** de **Filter**, desplegándose el árbol con todos los que están integrados.



Puede verse que los filtros de esta opción son de tipo no supervisado (*unsupervised*): son operaciones independientes del algoritmo análisis posterior, a diferencia de los filtros supervisados que se verán en la sección 1.9 de “selección de atributos”, que operan en conjunción con algoritmos de clasificación para analizar su efecto. Están agrupados según modifiquen los atributos resultantes o seleccionen un subconjunto de instancias (los filtros de atributos pueden verse como filtros "verticales" sobre la tabla de datos, y los filtros de instancias como filtros "horizontales"). Como puede verse, hay más de 30 posibilidades, de las que destacaremos únicamente algunas de las más frecuentes.

1.4.2.1. Filtros de atributos

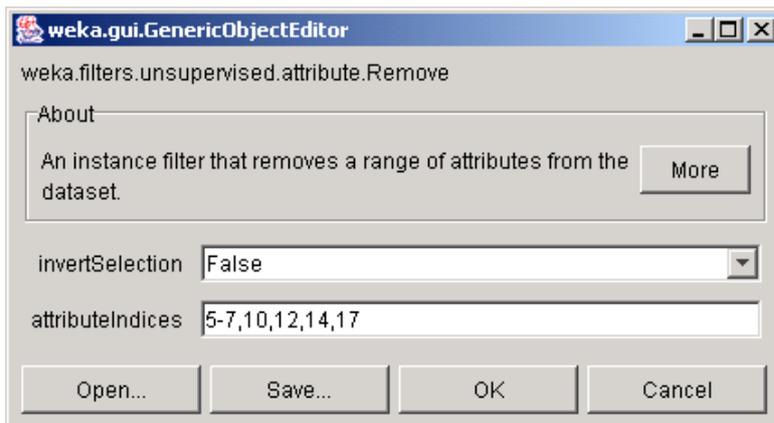
Vamos a indicar, de entre todas las posibilidades implementadas, la utilización de filtros para eliminar atributos, para discretizar atributos numéricos, y para añadir nuevos atributos con expresiones, por la frecuencia con la que se realizan estas operaciones.

Filtros de selección

Vamos a utilizar el filtro de atributos “*Remove*”, que permite eliminar una serie de atributos del conjunto de entrada. En primer lugar procedemos a seleccionarlo desde el árbol desplegado con el botón **Choose** de los filtros. A continuación lo configuraremos para determinar qué atributos queremos filtrar.

La configuración de un filtro sigue el esquema general de configuración de cualquier algoritmo integrado en WEKA. Una vez seleccionado el filtro específico con el botón **Choose**, aparece su nombre dentro del área de filtro (el lugar donde antes aparecía la palabra **None**). Se puede configurar sus parámetros haciendo clic sobre esta área, momento en el que aparece la ventana de configuración correspondiente a ese filtro particular. Si no se realiza esta operación se utilizarían los valores por defecto del filtro seleccionado.

Como primer filtro de selección, vamos a eliminar de los atributos de entrada todas las calificaciones parciales de la prueba y la calificación final, quedando como únicas calificaciones la nota de bachillerato y la calificación de la prueba. Por tanto tenemos que seleccionar los índices 5,6,7,10,12,14 y 17, indicándolo en el cuadro de configuración del filtro *Remove*:



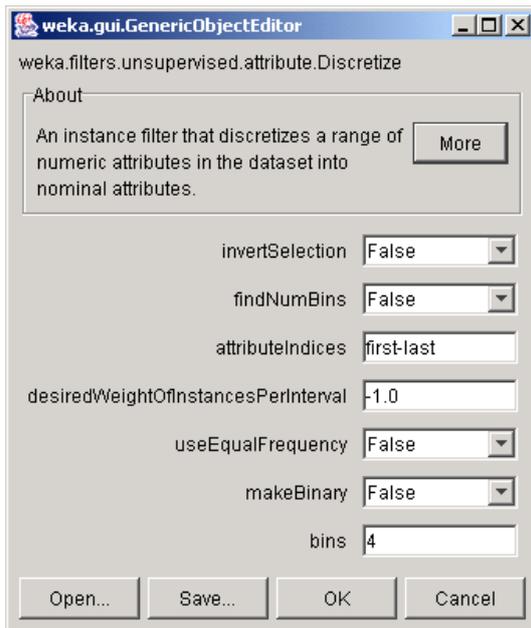
Como puede verse, en el conjunto de atributos a eliminar se pueden poner series de valores contiguos delimitados por guión (5-7) o valores sueltos entre comas (10,12,14,17). Además, puede usarse “first” y “last” para indicar el primer y último atributo, respectivamente. La opción **invertSelection** es útil cuando realmente queremos seleccionar un pequeño subconjunto de todos los atributos y eliminar el resto. **Open** y **Save** nos permiten guardar configuraciones de interés en archivos. El botón **More**, que aparece opcionalmente en algunos elementos de WEKA, muestra información de utilidad acerca de la configuración de los mismos. Estas convenciones para designar y seleccionar atributos, ayuda, y para guardar y cargar configuraciones específicas es común a otros elementos de WEKA.

Una vez configurado, al accionar el botón **Apply** del área de filtros se modifica el conjunto de datos (se filtra) y se genera una relación transformada. Esto se hace indicar en la descripción “Current Relation”, que pasa a ser la resultante de aplicar la operación correspondiente (esta información se puede ver con más nitidez en la ventana de log, que además nos indicará la cascada de filtros aplicados a la relación operativa). La relación transformada tras aplicar el filtro podría almacenarse en un nuevo fichero ARFF con el botón **Save**, dentro de la ventana **Preprocess**.

Filtros de discretización

Estos filtros son muy útiles cuando se trabaja con atributos numéricos, puesto que muchas herramientas de análisis requieren datos simbólicos, y por tanto se necesita aplicar esta transformación antes. También son necesarios cuando queremos hacer una clasificación sobre un atributo numérico, por ejemplo clasificar los alumnos aprobados y suspensos. Este filtrado transforma los atributos numéricos seleccionados en atributos simbólicos, con una serie de etiquetas resultantes de dividir la amplitud total del atributo en intervalos, con diferentes opciones para seleccionar los límites. Por defecto, se divide la amplitud del intervalo en tantas "cajas" como se indique en **bins** (por defecto 10), todas ellas de la misma amplitud.

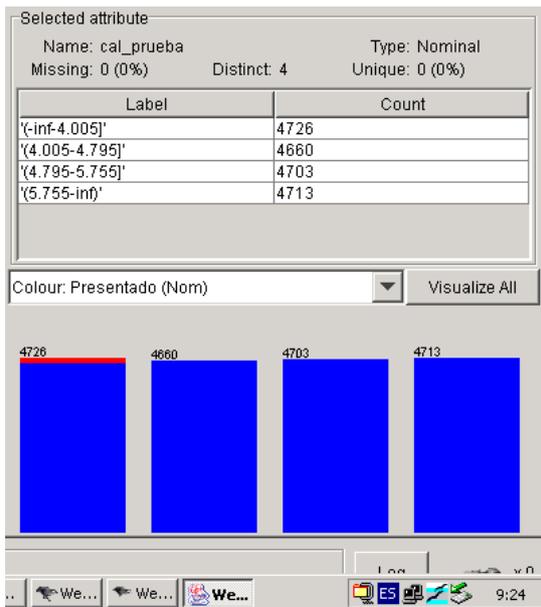
Por ejemplo, para discretizar las calificaciones numéricas en 4 categorías, todas de la misma amplitud, se configuraría así:



observe el resultado después de aplicar el filtro y los límites elegidos para cada atributo. En este caso se ha aplicado a todos los atributos numéricos con la misma configuración (los atributos seleccionados son first-last, no considerando los atributos que antes del filtrado no eran numéricos). Observe que la relación de trabajo ahora ("current relation") ahora es el resultado de aplicar en secuencia el filtro anterior y el actual.

A veces es más útil no fijar todas las cajas de la misma anchura sino forzar a una distribución uniforme de instancias por categoría, con la opción **useEqualFrequency**. La opción **findNumBins** permite optimizar el número de cajas (de la misma amplitud), con un criterio de clasificación de mínimo error en función de las etiquetas.

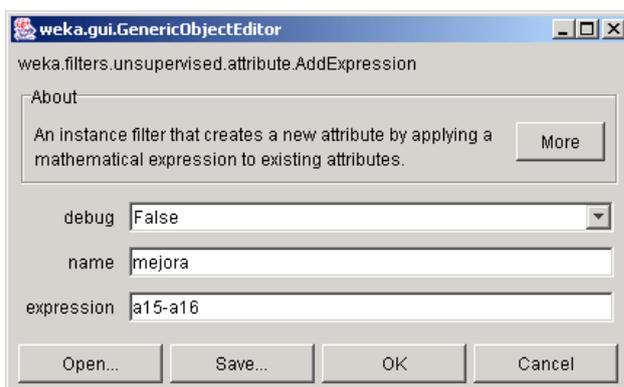
Haga una nueva discretización de la relación (eliminando el efecto del filtro anterior y dejando la relación original con el botón **Undo**) que divida las calificaciones en 4 intervalos de la misma frecuencia, lo que permite determinar los cuatro cuartiles (intervalos al 25%) de la calificación en la prueba: los intervalos delimitados por los valores {4, 4.8, 5.76}



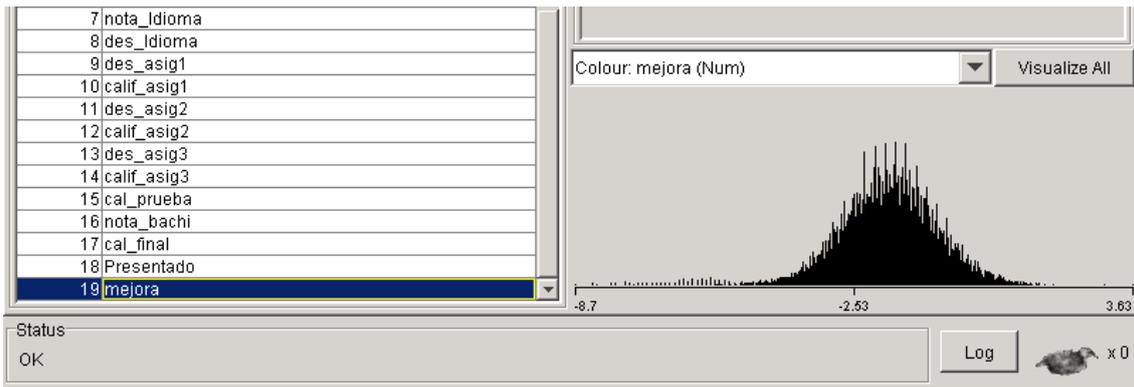
podemos ver que el 75% alcanza la nota de compensación (4). El 50% está entre 4 y 5.755, y el 25% restante a partir de 5.755.

Filtros de añadir expresiones

Muchas veces es interesante incluir nuevos atributos resultantes de aplicar expresiones a los existentes, lo que puede traer información de interés o formular cuestiones interesantes sobre los datos. Por ejemplo, vamos a añadir como atributo de interés la "mejora" sobre la nota de bachillerato, lo que puede servir para calificar el "éxito" en la prueba. Seleccionamos el filtro de atributos **AddExpression**, configurado para obtener la diferencia entre los atributos calificación en la prueba, y nota de bachillerato, en las posiciones 15 y 16:



después de aplicarlo aparece este atributo en la relación, sería el número 19, con el histograma indicado en la figura:

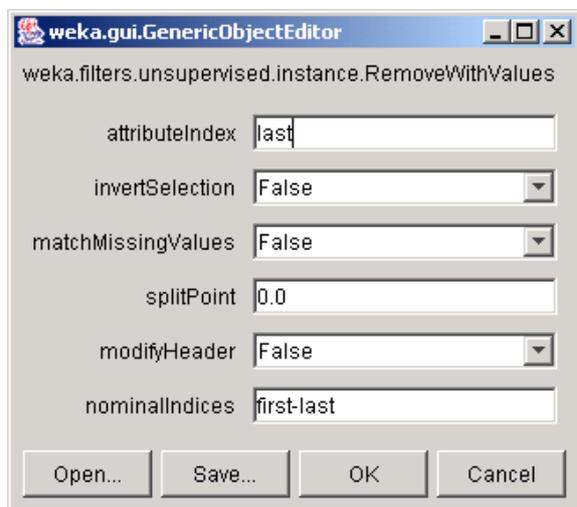


1.4.2.2. Filtros de instancias

De entre todas las posibilidades implementadas para filtros de selección de instancias (selección de rangos, muestreos, etc.), nos centraremos en la utilización de filtros para seleccionar instancias cuyos atributos cumplen determinadas condiciones.

Selección de instancias con condiciones sobre atributos

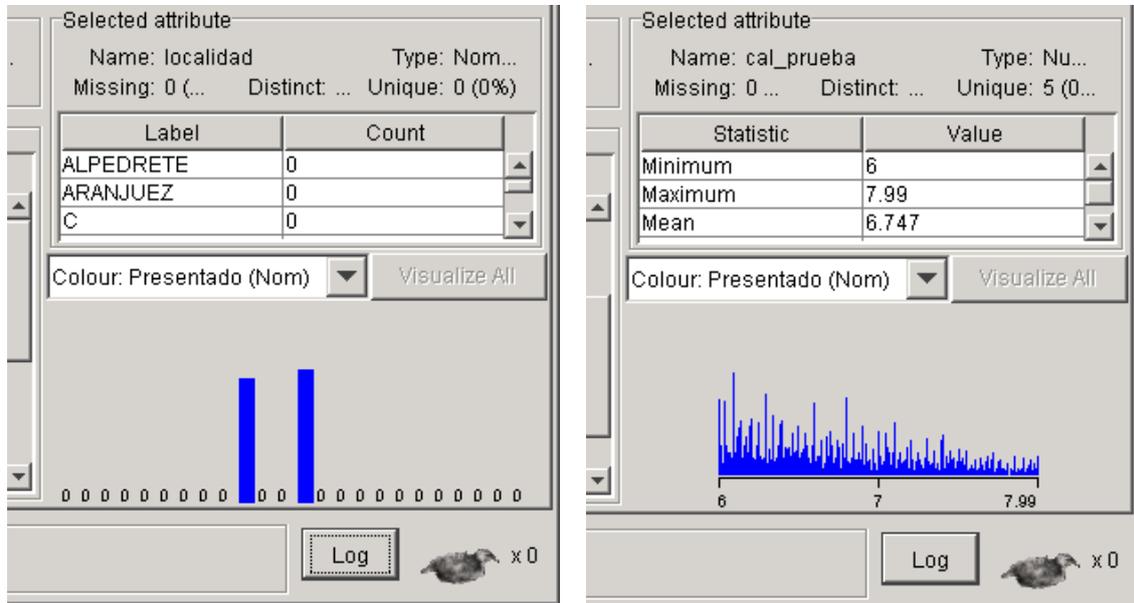
Vamos a utilizar el filtro **RemoveWithValues**, que elimina las instancias de acuerdo a condiciones definidas sobre uno de los atributos. Las opciones que aparecen en la ventana de configuración son las indicadas a continuación.



el atributo utilizado para filtrar se indica en "attributeIndex". Si es un atributo nominal, se indican los valores a filtrar en el último parámetro, "nominalIndices". Si es numérico, se filtran las instancias con un valor inferior al punto de corte, "splitPoint". Se puede invertir el criterio de filtrado mediante el campo "invertSelection".

Este filtro permite verificar una condición simple sobre un atributo. Sin embargo, es posible hacer un filtrado más complejo sobre varias condiciones aplicadas a uno o varios atributos sin más que aplicar en cascada varios filtros

A modo de ejemplo, utilice tres filtros de este tipo para seleccionar los alumnos de Getafe y Leganés con una calificación de la prueba entre 6.0 y 8.0. Compruebe el efecto de filtrado visualizando los histogramas de los atributos correspondientes (localidad y calificación en la prueba), tal y como se indica en la figura siguiente:

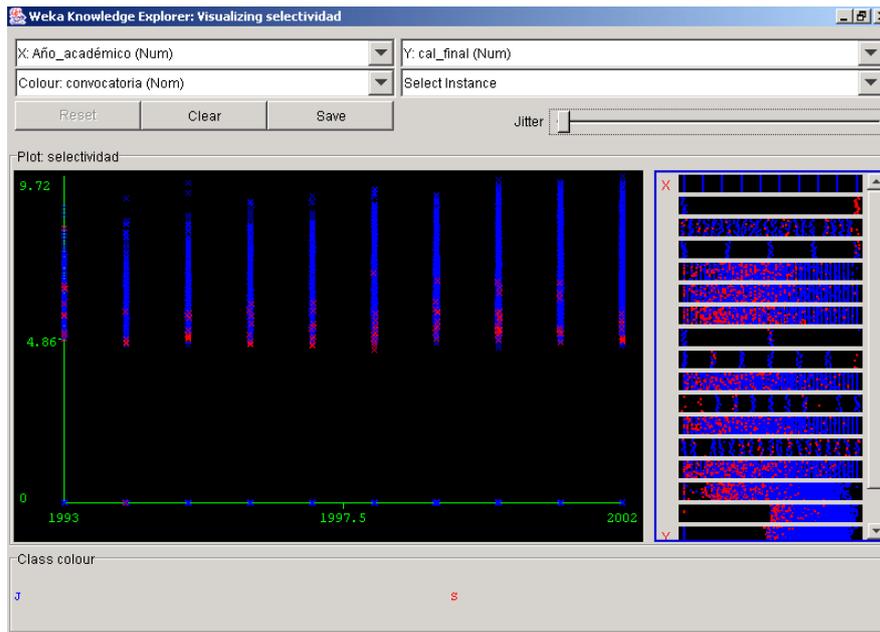


1.5. Visualización

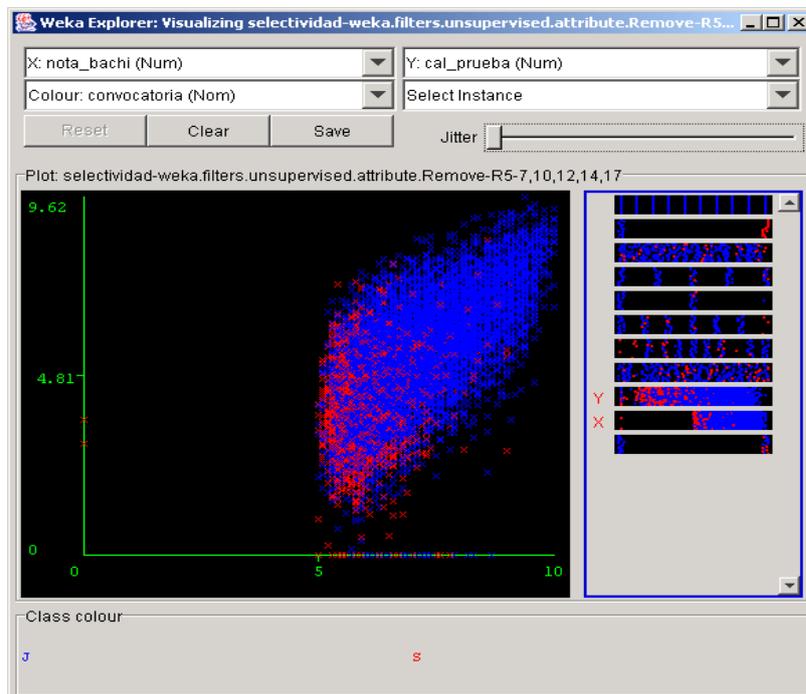
Una de las primeras etapas del análisis de datos puede ser el mero análisis visual de éstos, en ocasiones de gran utilidad para desvelar relaciones de interés utilizando nuestra capacidad para comprender imágenes. La herramienta de visualización de WEKA permite presentar gráficas 2D que relacionen pares de atributos, con la opción de utilizar además los colores para añadir información de un tercer atributo. Además, tiene incorporada una facilidad interactiva para seleccionar instancias con el ratón.

1.5.1. Representación 2D de los datos

Las instancias se pueden visualizar en gráficas 2D que relacionen pares de atributos. Al seleccionar la opción **Visualize** del *Explorer* aparecen todas los pares posibles de atributos en las coordenadas horizontal y vertical. La idea es que se selecciona la gráfica deseada para verla en detalle en una ventana nueva. En nuestro caso, aparecerán todas las combinaciones posibles de atributos. Como primer ejemplo vamos a visualizar el rango de calificaciones finales de los alumnos a lo largo de los años, poniendo la convocatoria (junio o septiembre) como color de la gráfica.



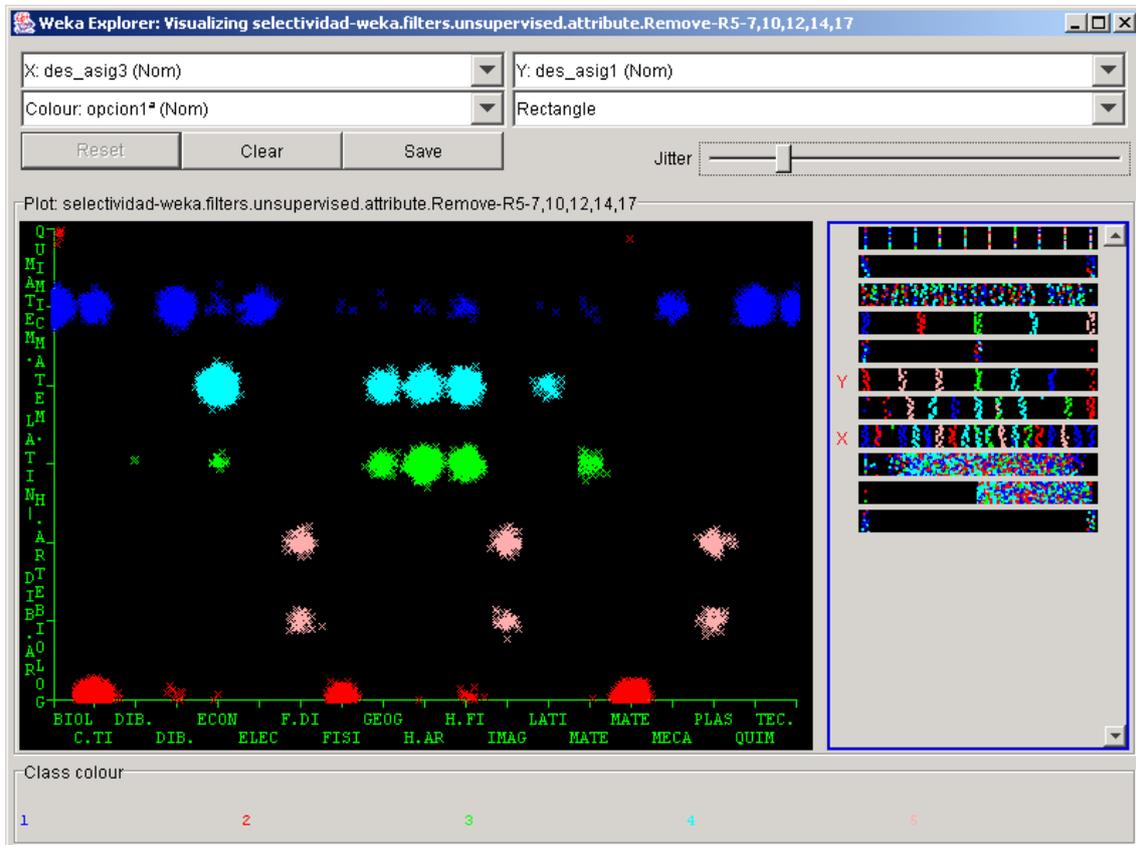
Vamos a visualizar ahora dos variables cuya relación es de gran interés, la calificación de la prueba en función de la nota de bachillerato, y tomando como color la convocatoria (junio o septiembre).

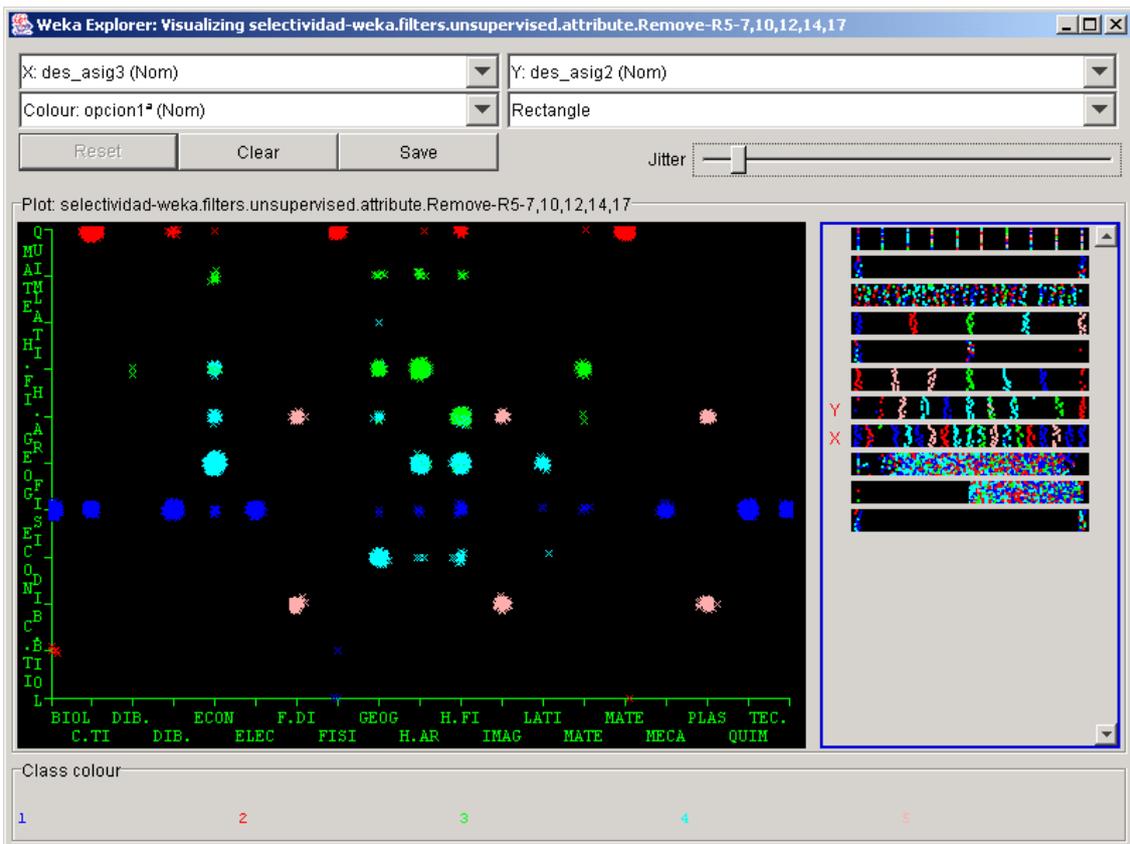


en esta gráfica podemos apreciar la relación entre ambas magnitudes, que si bien no es directa al menos define una cierta tendencia creciente, y como la convocatoria está bastante relacionada con ambas calificaciones.

Cuando lo que se relacionan son variables simbólicas, se presentan sus posibles valores a lo largo del eje. Sin embargo, en estos casos todas las instancias que comparten cada valor de un atributo simbólico pueden ocultarse

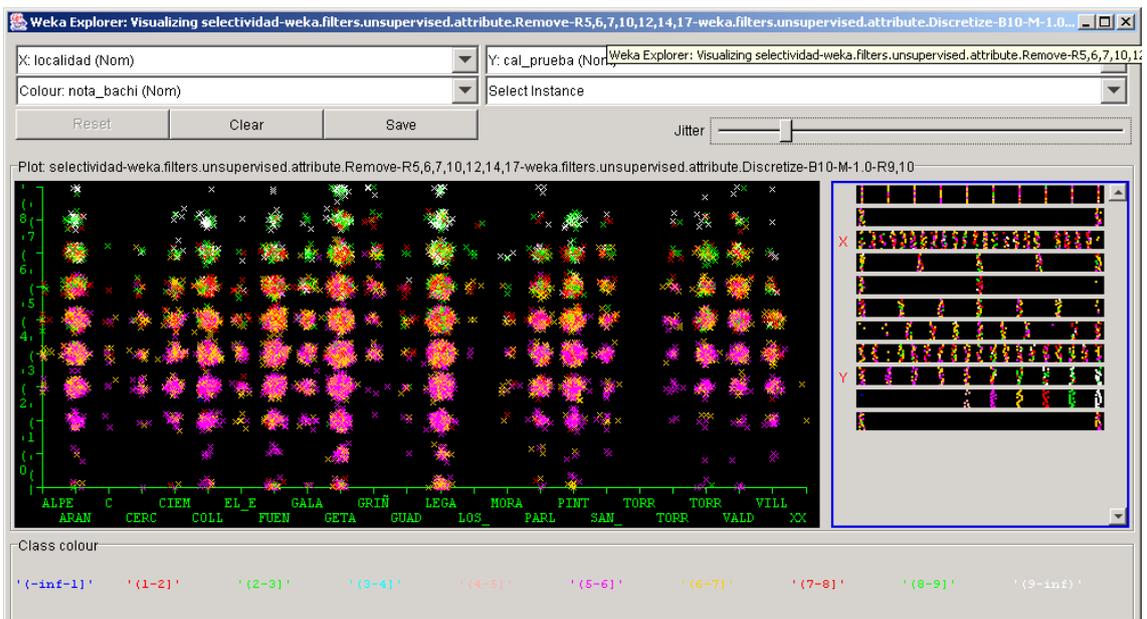
(serían un único punto en el plano), razón por la que se utiliza la facilidad de **Jitter**. Esta opción permite introducir un desplazamiento aleatorio (ruido) en las instancias, con objeto de poder visualizar todas aquellas que comparten un par de valores de atributos simbólicos, de manera que puede visualizarse la proporción de instancias que aparece en cada región. A modo de ejemplo se muestra a continuación la relación entre las tres asignaturas optativas, y con la opción cursada como color





puede verse una marcada relación entre las asignaturas opcionales, de manera que este gráfico ilustra qué tipo de asignaturas engloba cada una de las cinco posibles opciones cursadas.

Se sugiere preparar el siguiente gráfico, que relaciona la calificación obtenida en la prueba con la localidad de origen y la nota de bachillerato, estando las calificaciones discretizadas en intervalos de amplitud 2



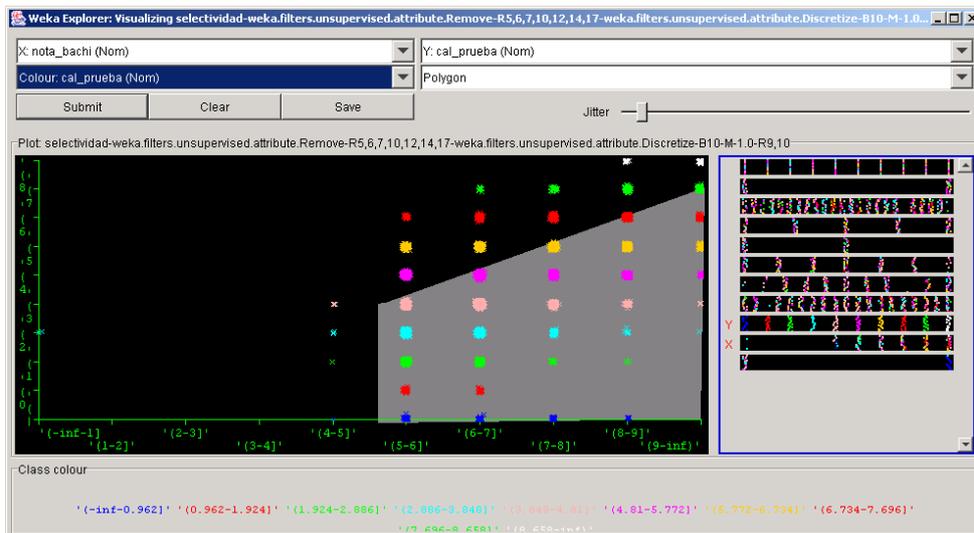
Aquí el color trae más información, pues indica en cada intervalo de calificaciones de la prueba, la calificación en bachillerato, lo que permite ilustrar la "satisfacción" con la calificación en la prueba o resultados no esperados, además distribuido por localidades.

1.5.2. Filtrado “gráfico” de los datos

WEKA permite también realizar filtros de selección de instancias sobre los propios gráficos, con una interacción a través del ratón para aislar los grupos de instancias cuyos atributos cumplen determinadas condiciones. Esta facilidad permite realizar filtrados de instancias de modo interactivo y más intuitivo que los filtros indicados en la sección 1.4.2.2. Las opciones que existen son:

- Selección de instancias con un valor determinado (hacer clic sobre la posición en el gráfico)
- Selección con un rectángulo de un subconjunto de combinaciones (comenzando por el vértice superior izquierdo) (**Rectangle**)
- Selección con un polígono cerrado de un subconjunto (**Polygon**)
- Selección con una línea abierta de frontera (**Polyline**)

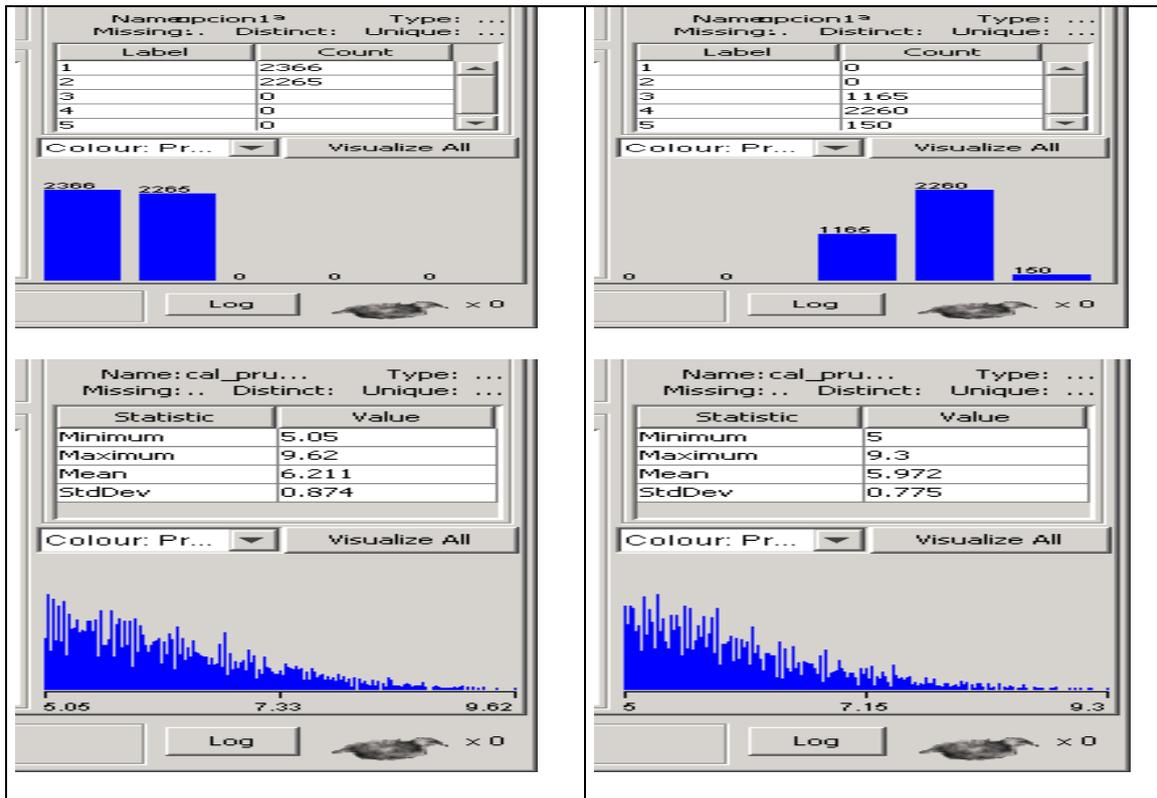
Por ejemplo, a continuación se indica la selección de alumnos que obtuvieron una calificación por debajo de sus expectativas (calificación en la prueba inferior a su nota en el bachillerato), con la opción **Polygon**.



Una vez realizada la selección, la opción **Submit** permite eliminar el resto de instancias, y **Save** almacenarlas en un fichero. **Reset** devuelve la relación a su estado original.

Utilice estas facilidades gráficas para hacer subconjuntos de los datos con los alumnos aprobados de las opciones 1 y 2 frente a los de las opciones 3, 4 y 5.

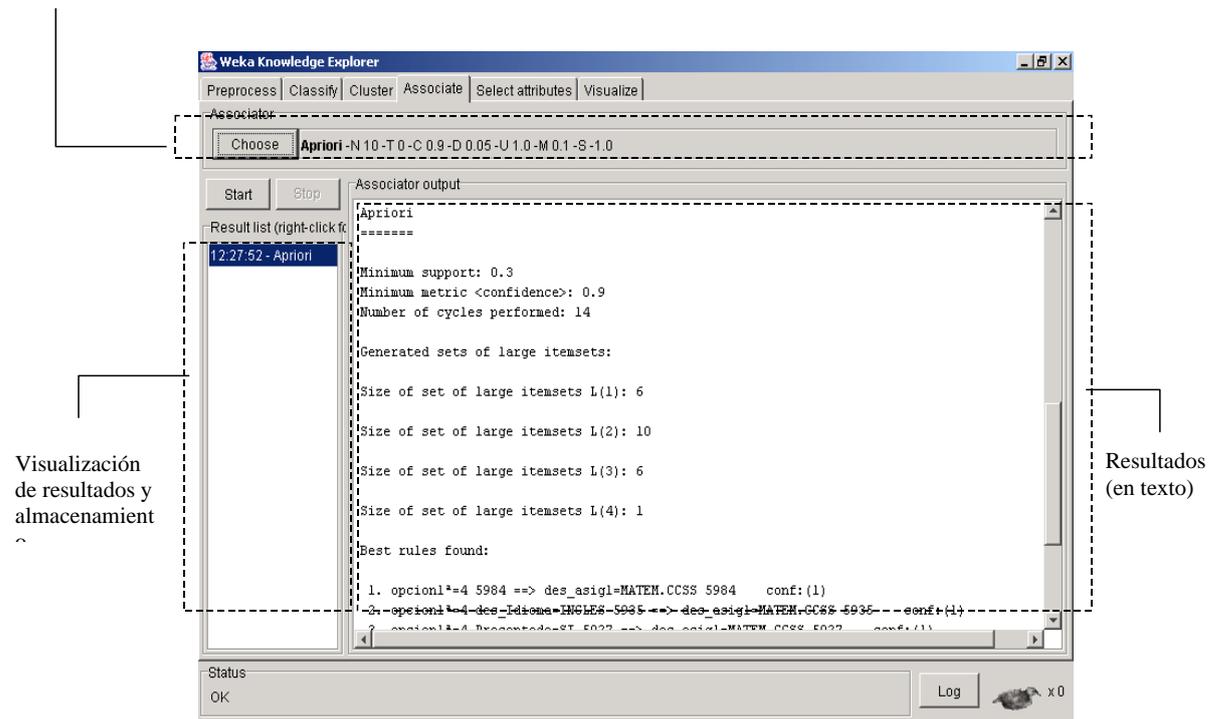
Salve las relaciones filtradas para a continuación cargarlas y mostrar los histogramas, que aparecerán como se indica en la figura siguiente.



1.6. Asociación

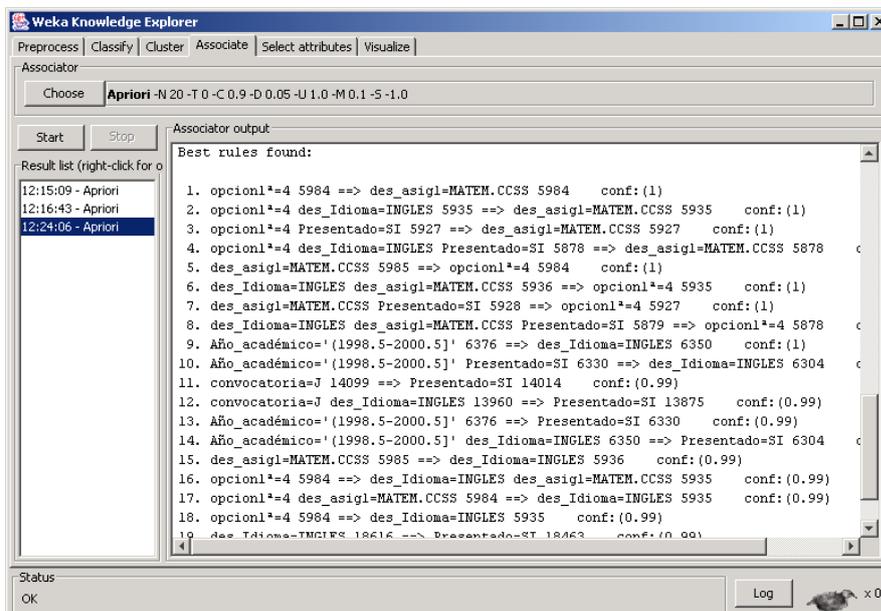
Los algoritmos de asociación permiten la búsqueda automática de reglas que relacionan conjuntos de atributos entre sí. Son algoritmos no supervisados, en el sentido de que no existen relaciones conocidas a priori con las que contrastar la validez de los resultados, sino que se evalúa si esas reglas son estadísticamente significativas. La ventana de Asociación (**Associate** en el Explorer), tiene los siguiente elementos:

Selección y configuración del algoritmo de asociación

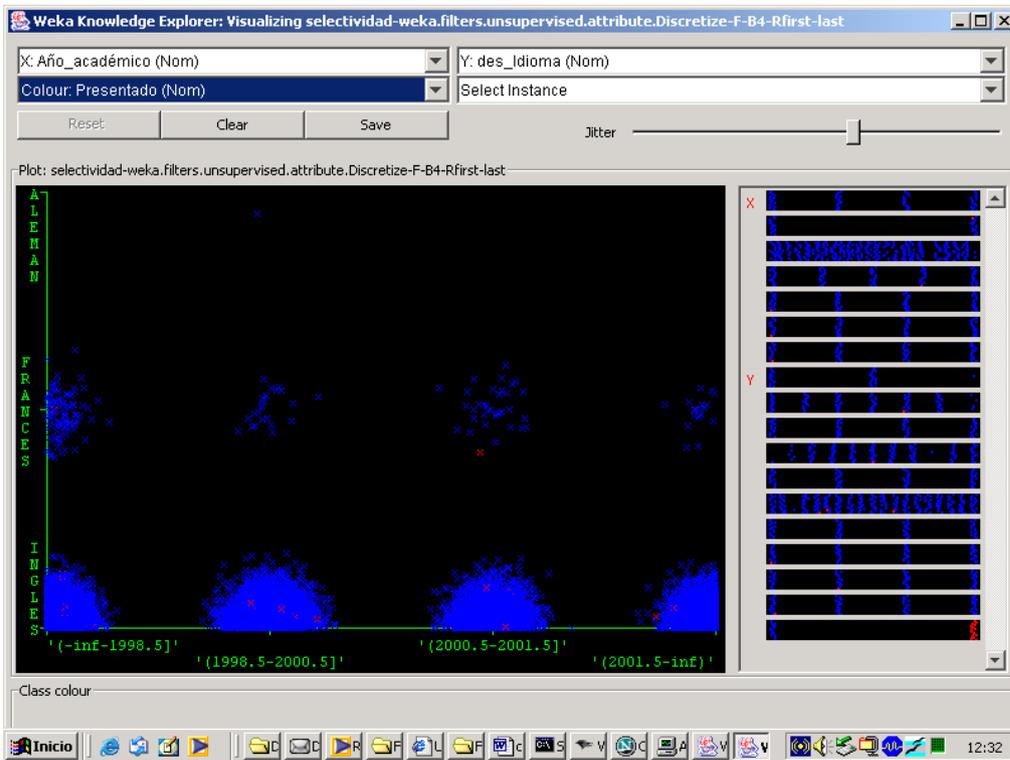


El principal algoritmo de asociación implementado en WEKA es el algoritmo "Apriori". Este algoritmo únicamente puede buscar reglas entre atributos simbólicos, razón por la que se requiere haber discretizado todos los atributos numéricos.

Por simplicidad, vamos a aplicar un filtro de discretización de todos los atributos numéricos en cuatro intervalos de la misma frecuencia para explorar las relaciones más significativas. El algoritmo lo ejecutamos con sus parámetros por defecto.



las reglas que aparecen aportan poca información. Aparecen en primer lugar las relaciones triviales entre asignaturas y opciones, así como las que relacionan suspensos en la prueba y en la calificación final. En cuanto a las que relacionan alumnos presentados con idioma seleccionado son debidas a la fuerte descompensación en el idioma seleccionado. La abrumadora mayoría de los presentados a la prueba de idioma seleccionaron el inglés, como indica la figura siguiente:



Con objeto de buscar relaciones no conocidas, se filtrarán ahora todos los atributos relacionados con descriptores de asignaturas y calificaciones parciales, quedando únicamente los atributos:

```
Año_académico
convocatoria
localidad
opcion1a
cal_prueba
nota_bachi
```

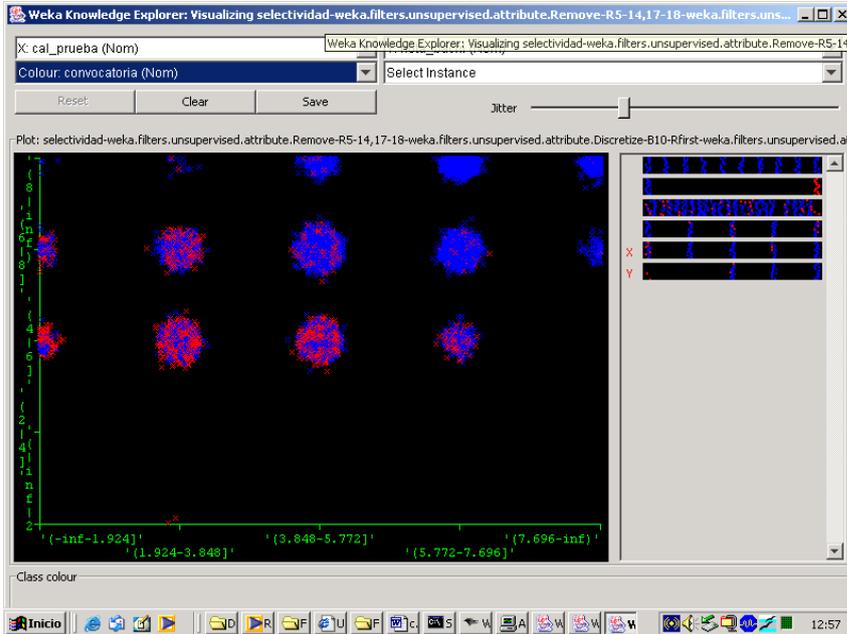
En este caso, las reglas más significativas son:

1. nota_bachi='(8-inf)' 2129 ==> convocatoria=J 2105 conf:(0.99)
2. cal_prueba='(5.772-7.696]' nota_bachi='(6-8]' 2521 ==>

convocatoria=J 2402 conf:(0.95)
3. cal_prueba='(5.772-7.696]' 4216 ==>

convocatoria=J 3997 conf:(0.95)

estas reglas aportan información no tan trivial: el 99% de alumnos con nota superior a 8 se presentan a la convocatoria de Junio, así el 95% de los alumnos con calificación en la prueba entre 5.772 y 7.



es significativo ver que no aparece ninguna relación importante entre las calificaciones, localidad y año de la convocatoria. También es destacado ver la ausencia de efecto de la opción cursada.

Si preparamos los datos para dejar sólo cinco atributos,

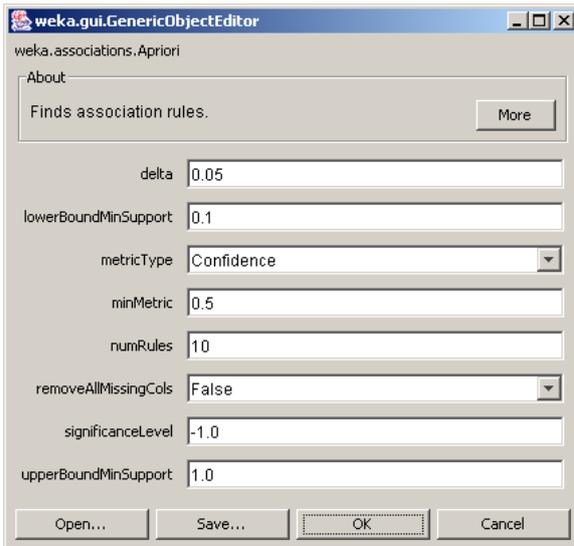
```
Año_académico
convocatoria
localidad
opcion1ª
cal_final,
```

con el último discretizado en dos grupos iguales (hasta 5.85 y 5.85 hasta 10), tenemos que de nuevo las reglas más significativas relacionan convocatoria con calificación, pero ahora entran en juego opciones y localidades, si bien bajando la precisión de las reglas:

1. `opcion1ª=1 cal_final='(5.685-inf)' 2810 ==>`
`convocatoria=J 2615 conf:(0.93)`
2. `localidad=LEGANES cal_final='(5.685-inf)' 2514 ==>`
`convocatoria=J 2315 conf:(0.92)`
3. `Año_académico='(1998.4-2000.2]' cal_final='(5.685-inf)' 3175 ==>`
`convocatoria=J 2890 conf:(0.91)`
4. `cal_final='(5.685-inf)' 9397 ==>`
`convocatoria=J 8549 conf:(0.91)`
5. `opcion1ª=4 cal_final='(5.685-inf)' 2594 ==>`
`convocatoria=J 2358 conf:(0.91)`
6. `Año_académico='(2000.2-inf)' cal_final='(5.685-inf)' 3726 ==>`

```
convocatoria=J 3376    conf:(0.91)
7. localidad=GETAFE cal_final='(5.685-inf)' 2156 ==>
convocatoria=J 1951    conf:(0.9)
```

Al filtrar la convocatoria, que nos origina relaciones bastante evidentes, tendremos las reglas más significativas entre localidad, año, calificación y opción. Como podemos ver, al lanzar el algoritmo con los parámetros por defecto no aparece ninguna regla. Esto es debido a que se forzó como umbral mínimo aceptable para una regla el 90%. Vamos a bajar ahora este parámetro hasta el 50%:



Best rules found:

```
1. opcion1^a=4 5984 ==> cal_final='(-inf-5.685]' 3390    conf:(0.57)
2. opcion1^a=1 5131 ==> cal_final='(5.685-inf)' 2810    conf:(0.55)
3. Año_académico='(2000.2-inf)' 7049 ==>
   cal_final='(5.685-inf)' 3726    conf:(0.53)
4. opcion1^a=2 4877 ==> cal_final='(5.685-inf)' 2575    conf:(0.53)
5. localidad=GETAFE 4464 ==>
   cal_final='(-inf-5.685]' 2308    conf:(0.52)
6. localidad=LEGANES 4926 ==>
   cal_final='(5.685-inf)' 2514    conf:(0.51)
7. Año_académico='(1998.4-2000.2]' 6376 ==>
   cal_final='(-inf-5.685]' 3201    conf:(0.5)
```

Por tanto, forzando los términos, tenemos que los estudiantes de las 2 primeras opciones tienen mayor probabilidad de aprobar la prueba, así como los estudiantes de la localidad de Leganés. Los estudiantes de Getafe tienen una probabilidad superior de obtener una calificación inferior. Hay que destacar que estas reglas rozan el umbral del 50%, pero han sido seleccionadas como las más significativas de todas las posibles. También hay que considerar que si aparecen estas dos localidades en primer lugar es simplemente por su mayor volumen de datos, lo que otorga una significatividad superior en las relaciones encontradas. Si se consulta la bibliografía, el primer criterio de selección de reglas del algoritmo "A priori" es la precisión o confianza, dada por el porcentaje de veces que instancias que cumplen el antecedente cumplen el consecuente, pero el segundo es el soporte, dado por el número de instancias

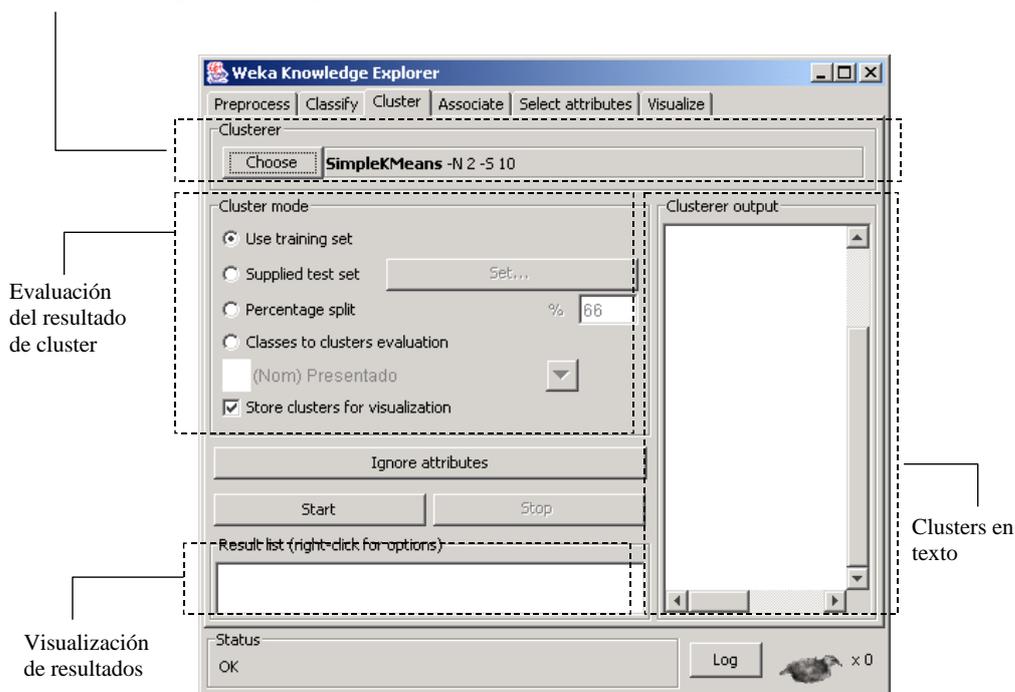
sobre las que es aplicable la regla. En todo caso, son reglas de muy baja precisión y que habría que considerar simplemente como ciertas tendencias.

1.7. Agrupamiento

La opción **Cluster** del *Experimenter* nos permite aplicar algoritmos de agrupamiento de instancias a nuestros datos. Estos algoritmos buscan grupos de instancias con características "similares", según un criterio de comparación entre valores de atributos de las instancias definidos en los algoritmos.

El mecanismo de selección, configuración y ejecución es similar a otros elementos: primero se selecciona el algoritmo con **Choose**, se ajustan sus parámetros seleccionando sobre el área donde aparece, y se después se ejecuta. El área de agrupamiento del Explorer presenta los siguientes elementos de configuración:

Selección y configuración del algoritmo



Una vez que se ha realizado la selección y configuración del algoritmo, se puede pulsar el botón **Start**, que hará que se aplique sobre la relación de trabajo. Los resultados se presentarán en la ventana de texto de la parte derecha. Además, la ventana izquierda permite listar todos los algoritmos y resultados que se hayan ejecutado en la sesión actual. Al seleccionarlos en esta lista de visualización se presentan en la ventana de texto a la derecha, y además se permite abrir ventanas gráficas de visualización con un menú contextual que aparece al pulsar el botón derecho sobre el resultado seleccionado. Por último, en esta opción de Agrupamiento aparecen las siguientes opciones adicionales en la pantalla.

Ignorar atributos

La opción **Ignoring Attributes** permite sacar fuera atributos que no interesa considerar para el agrupamiento, de manera que el análisis de parecido entre instancias no considera los atributos seleccionados. Al accionar esta opción aparecen todos los atributos disponibles. Se pueden seleccionar con el botón izquierdo sobre un atributo específico, o seleccionar grupos usando SHIFT para un grupo de atributos contiguos y CONTROL para grupos de atributos sueltos.

Evaluación

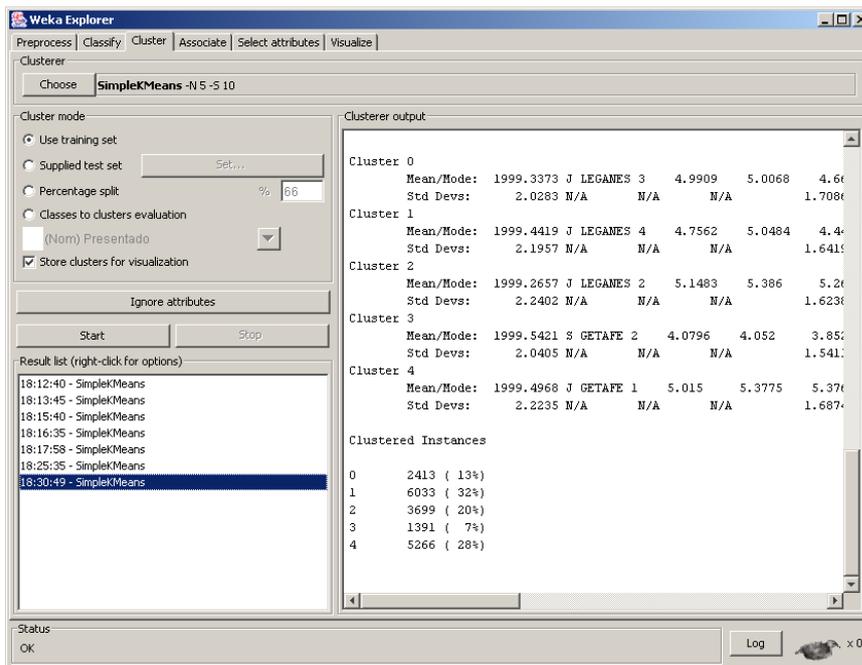
La opción **Cluster Mode** permite elegir como evaluar los resultados del agrupamiento. Lo más simple es utilizar el propio conjunto de entrenamiento, **Use training set**, que indica que porcentaje de instancias se van a cada grupo. El resto de opciones realizan un entrenamiento con un conjunto, sobre el que construyen los clusters y a continuación aplican estos clusters para clasificar un conjunto independiente que puede proporcionarse aparte (**Supplied test**), o ser un porcentaje del conjunto de entrada (**Percentage split**). Existe también la opción de comparar los clusters con un atributo de clasificación (**Classes to clusters evaluation**) que no se considera en la construcción de los clusters. Nosotros nos centraremos únicamente en la primera opción, dejando el resto de opciones de evaluación para más adelante, cuando llegemos a los algoritmos de clasificación.

Finalmente, el cuadro opcional de almacenamiento de instancias, **Store clusters for visualization**, es muy útil para después analizar los resultados gráficamente.

1.7.1. Agrupamiento numérico

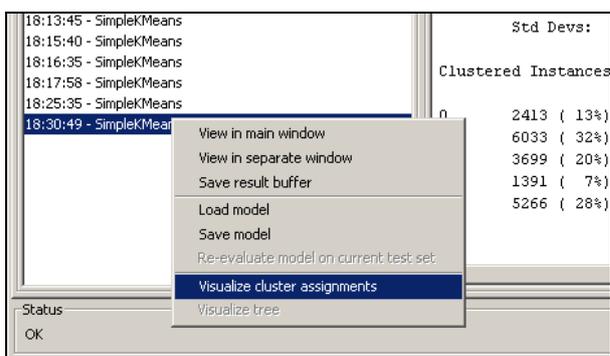
En primer lugar utilizaremos el algoritmo de agrupamiento K-medias, por ser uno de los más veloces y eficientes, si bien uno de los más limitados. Este algoritmo precisa únicamente del número de categorías similares en las que queremos dividir el conjunto de datos. Suele ser de interés repetir la ejecución del algoritmo K-medias con diferentes semillas de inicialización, dada la notable dependencia del arranque cuando no está clara la solución que mejor divide el conjunto de instancias.

En nuestro ejemplo, vamos a comprobar si el atributo “opción” divide naturalmente a los alumnos en grupos similares, para lo que seleccionamos el algoritmo **SimpleKMeans** con parámetro **numClusters** con valor 5. Los resultados aparecen en la ventana de texto derecha:

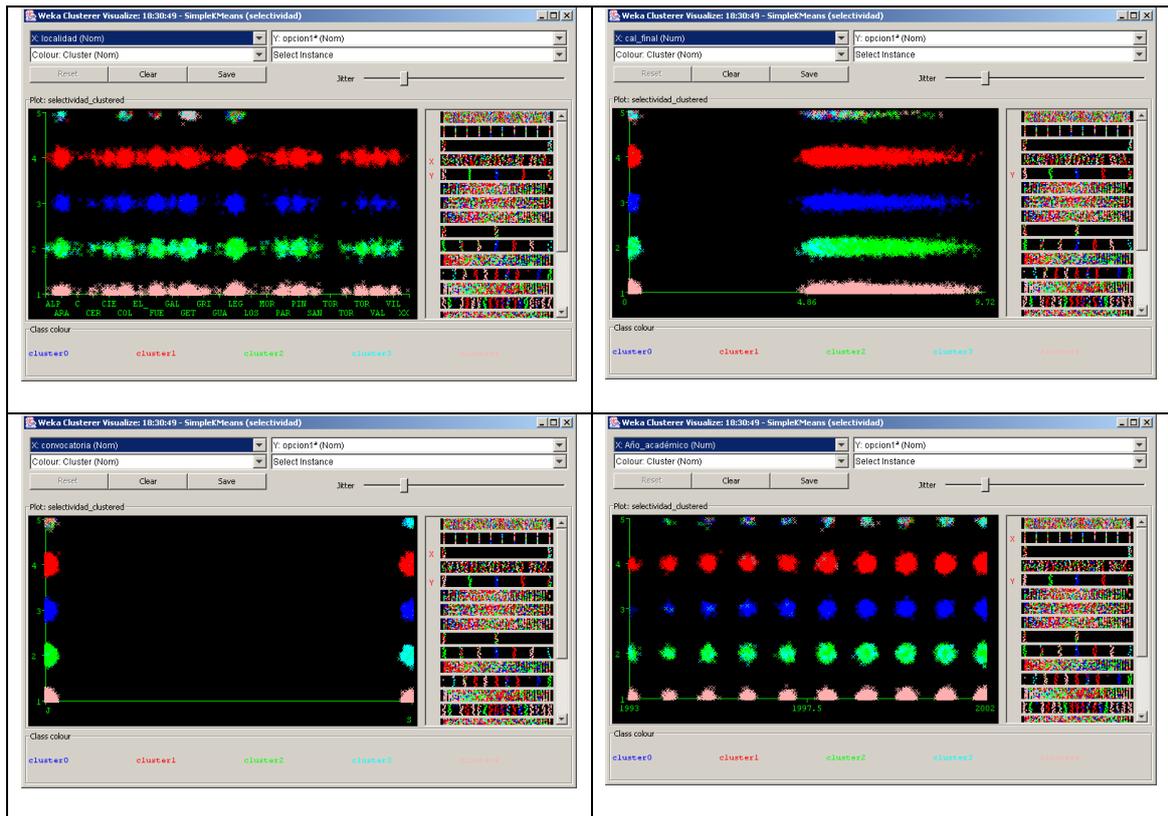


Nos aparecen los 5 grupos de ejemplos más similares, y sus centroides (promedios para atributos numéricos, y valores más repetidos en cada grupo para atributos simbólicos).

En este caso es de interés analizar gráficamente como se distribuyen diferentes valores de los atributos en los grupos generados. Para ello basta pulsar con botón derecho del ratón sobre el cuadro de resultados, y seleccionar la opción **visualizeClusterAssignments**

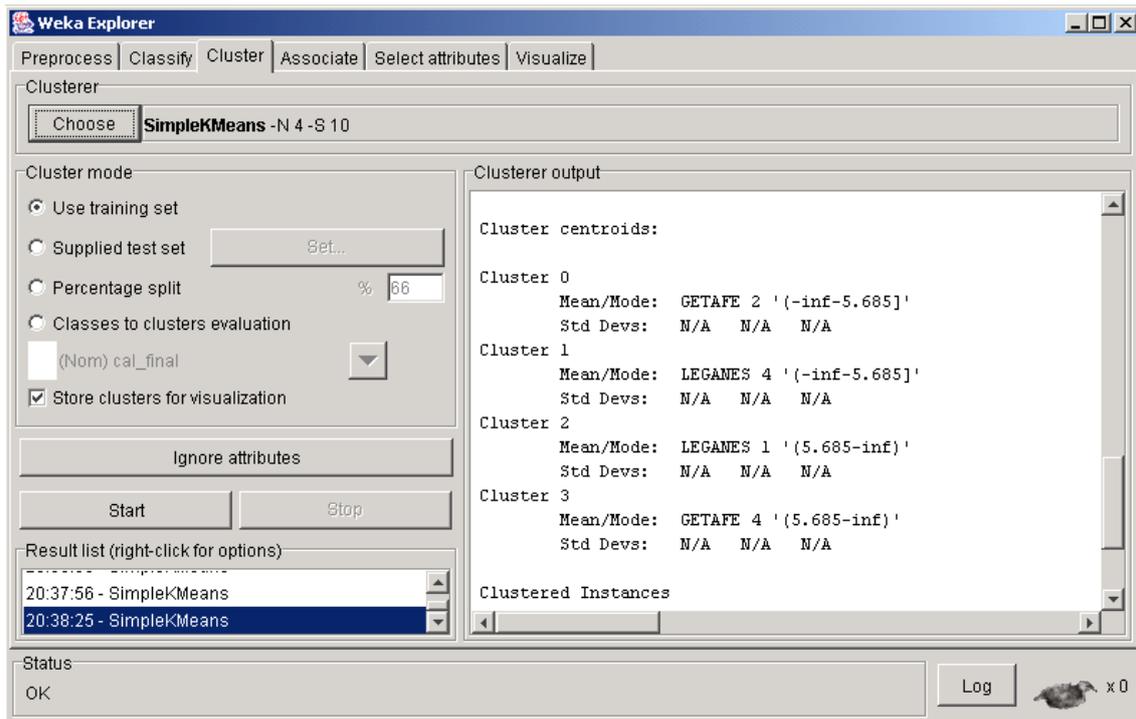


Si seleccionamos combinaciones del atributo opción con localidad, nota o convocatoria podemos ver la distribución de grupos:

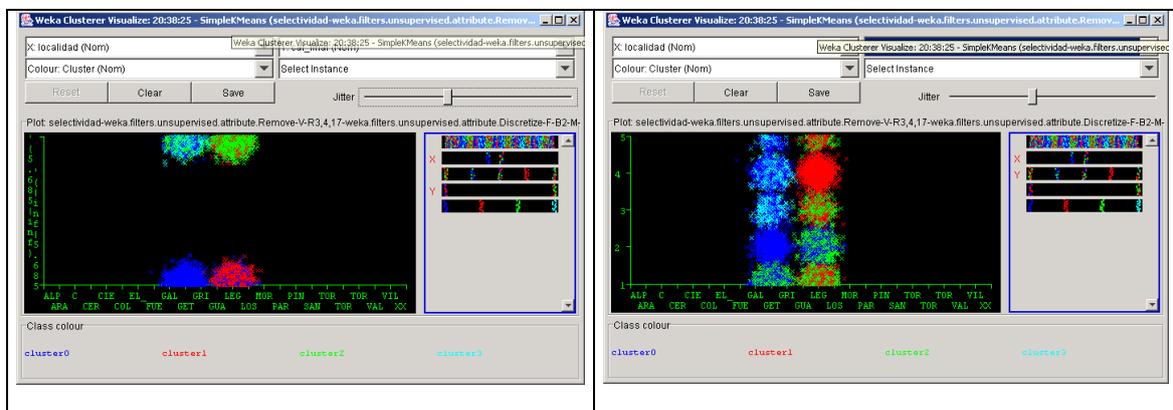


A la vista de estos gráficos podemos concluir que el “parecido” entre casos viene dado fundamentalmente por las opciones seleccionadas. Los clusters 0, 1 y 4 se corresponden con las opciones 3, 4 y 1, mientras que los clusters 2 y 3 representan la opción 3 en las convocatorias de junio y septiembre.

Aprovechando esta posibilidad de buscar grupos de semejanzas, podríamos hacer un análisis más particularizado a las dos localidades mayores, Leganés y Getafe, buscando qué opciones y calificaciones aparecen con más frecuencia. Vamos a preparar los datos con filtros de modo que tengamos únicamente tres atributos: localidad, opción, y calificación final. Además, discretizamos las calificaciones en dos grupos de la misma frecuencia (estudiantes con mayor y menor éxito), y únicamente nos quedamos con los alumnos de Leganés y Getafe. Utilizaremos para ello los filtros adecuados. A continuación aplicamos el algoritmo K-medias con 4 grupos.



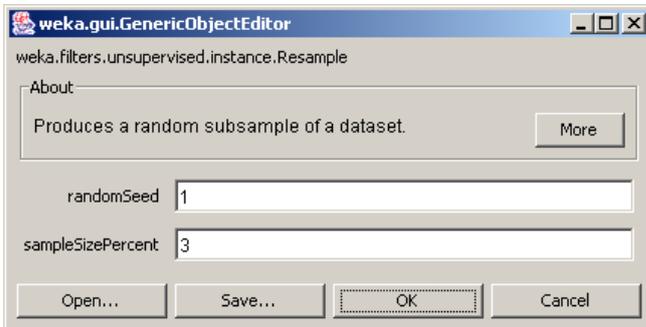
vemos que los grupos nos muestran la presencia de buenos alumnos en Getafe en la opción 4, y buenos alumnos en Leganés en la opción 1, siempre considerando estas conclusiones como tendencias en promedio. Gráficamente vemos la distribución de clusters por combinaciones de atributos:



Si consideramos que en Leganés hay escuelas de ingeniería, y en Getafe facultades de Humanidades, podríamos concluir que podría ser achacable al impacto de la universidad en la zona.

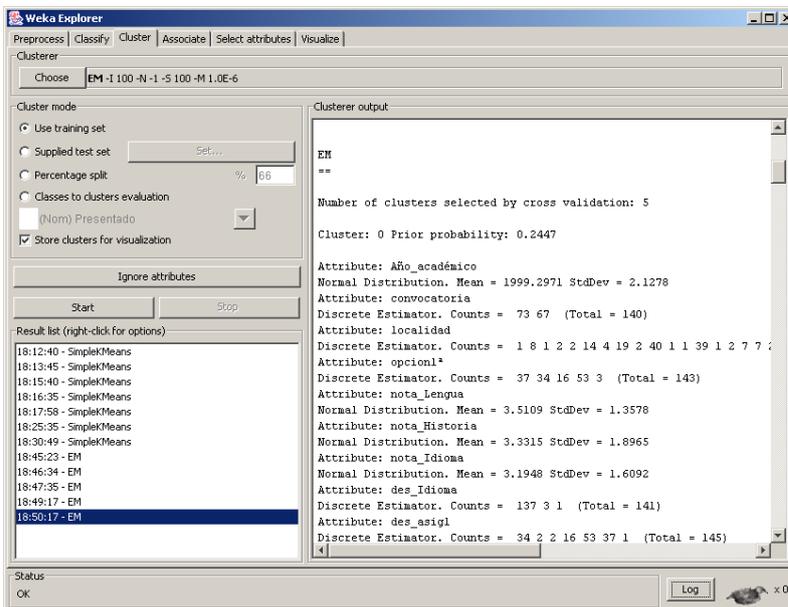
El algoritmo EM proviene de la estadística y es bastante más elaborado que el K-medias, con el coste de que requiere muchas más operaciones, y es apropiado cuando sabemos que los datos tienen una variabilidad estadística de modelo conocido. Dada esta mayor complejidad, y el notable volumen del

fichero de datos, primero aplicaremos un filtro de instancias al 3% para dejar un número de 500 instancias aproximadamente. Para esto último iremos al preprocesado y aplicamos un filtro de instancias, el filtro **Resample**, con factor de reducción al 3%:

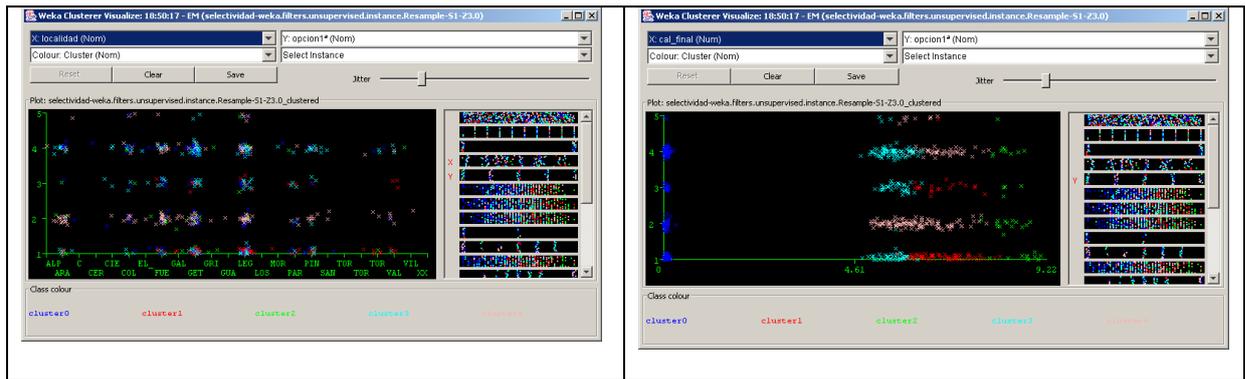


Una ventaja adicional del algoritmo de clustering EM es que permite además buscar el número de grupos más apropiado, para lo cual basta indicar a -1 el número de clusters a formar, que es la opción que viene por defecto. Esto se interpreta como dejar el parámetro del número de clusters como un valor a optimizar por el propio algoritmo.

Tras su aplicación, este algoritmo determina que hay cinco clusters significativos en la muestra de 500 alumnos, y a continuación indica los centroides de cada grupo:



Al igual que antes, es interesante analizar el resultado del agrupamiento sobre diferentes combinaciones de atributos, haciendo uso de la facilidad **visualizeClusterAssignments**



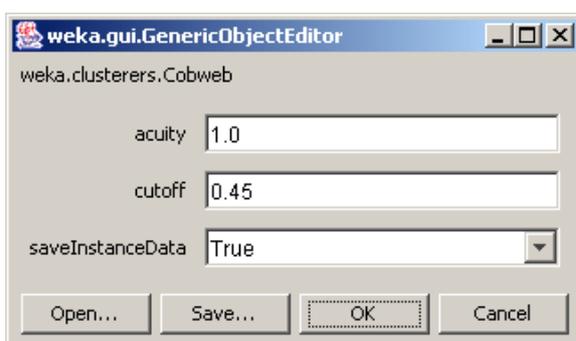
Por tanto podría concluirse que para este segundo algoritmo de agrupamiento por criterios estadísticos y no de distancias entre vectores de atributos, predomina el agrupamiento de los alumnos básicamente por tramos de calificaciones, independientemente de la opción, mientras que en el anterior pesaba más el perfil de asignaturas cursado que las calificaciones.

Esta disparidad sirve para ilustrar la problemática de la decisión del criterio de “parecido” entre instancias para realizar el agrupamiento.

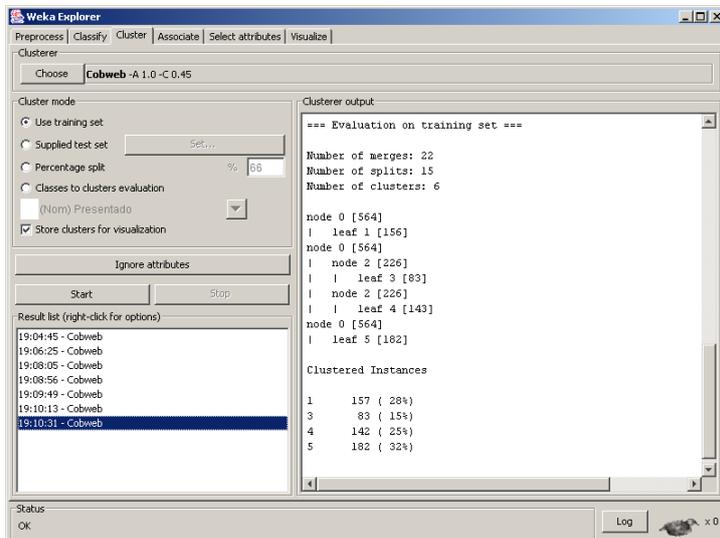
1.7.2. Agrupamiento simbólico

Finalmente, como alternativa a los algoritmos de agrupamiento anteriores, el agrupamiento simbólico tiene la ventaja de efectuar un análisis cualitativo que construye categorías jerárquicas para organizar los datos. Estas categorías se forman con un criterio probabilístico de "utilidad", llegando a las que permiten homogeneidad de los valores de los atributos dentro de cada una y al mismo tiempo una separación entre categorías dadas por los atributos, propagándose estas características en un árbol de conceptos.

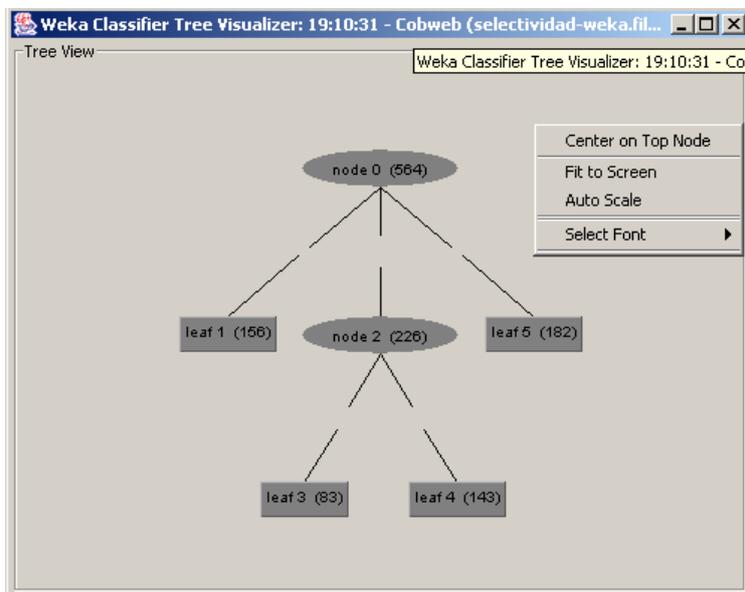
Si aplicamos el algoritmo cobweb con los parámetros por defecto sobre la muestra reducida de instancias (dada la complejidad del algoritmo), el árbol generado llega hasta 800 nodos. Vamos a modificar el parámetro **cut-off**, que permite poner condiciones más restrictivas a la creación de nuevas categorías en un nivel y subcategorías. Con los parámetros siguientes se llega a un árbol muy manejable:



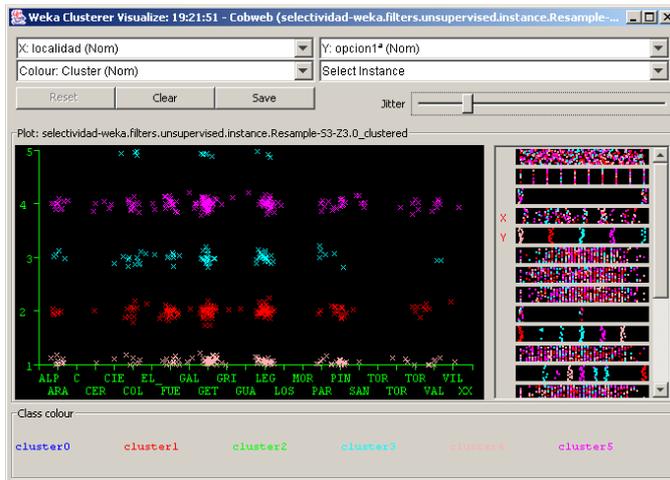
la opción **saveInstanceData** es de gran utilidad para después analizar la distribución de valores de atributos entre las instancias de cada parte del árbol de agrupamiento. Una vez ejecutado Cobweb, genera un resultado como el siguiente:



hay 3 grupos en un primer nivel, y el segundo se subdivide en otros dos. De nuevo activando el botón derecho sobre la ventana de resultados, ahora podemos visualizar el árbol gráficamente:



las opciones de visualización aparecen al pulsar el botón derecho en el fondo de la figura. Se pueden visualizar las instancias que van a cada nodo sin más que pulsar el botón derecho sobre él. Si nos fijamos en como quedan distribuidas las instancias por clusters, con la opción **visualizeClusterAssignments**, llegamos a la figura:



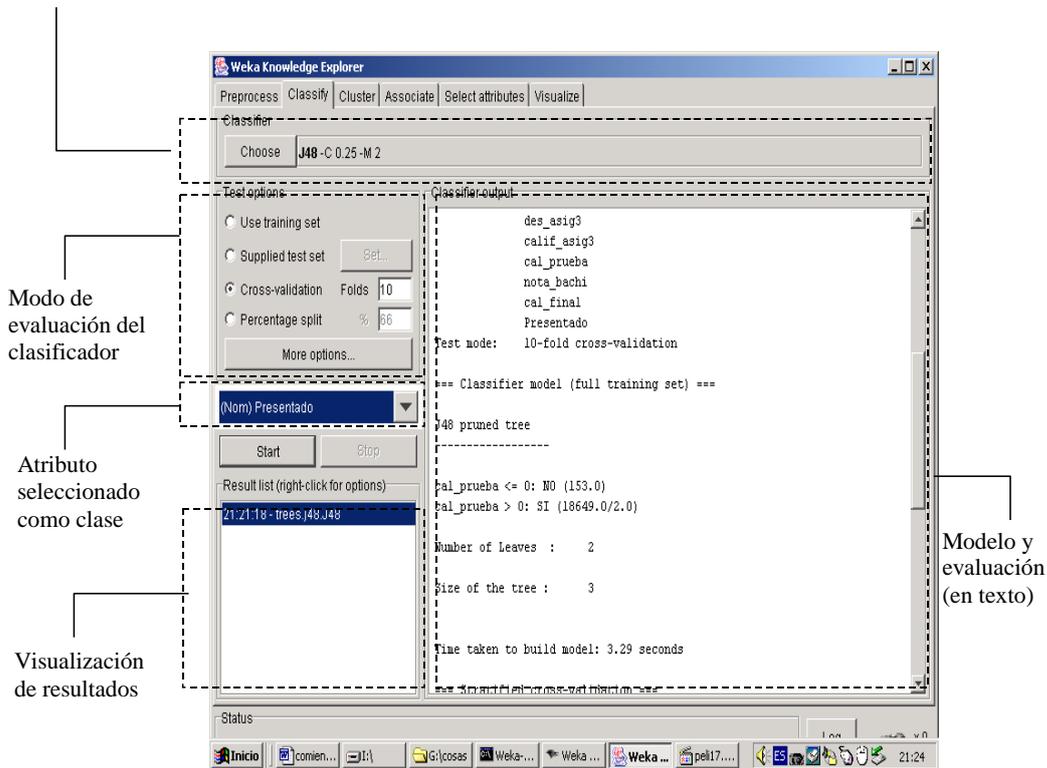
por tanto, vemos que de nuevo vuelve a pesar la opción como criterio de agrupamiento. Los nodos hoja 1, 3, 4 y 5 se corresponden con las opciones cursadas 2, 3, 1 y 4 respectivamente. En un primer nivel hay tres grupos, uno para la opción 2, otro para la opción 4 y otro que une las opciones 1 y 3. Este último se subdivide en dos grupos que se corresponden con ambas opciones.

1.8. Clasificación

Finalmente, en esta sección abordamos el problema de la clasificación, que es el más frecuente en la práctica. En ocasiones, el problema de clasificación se formula como un refinamiento en el análisis, una vez que se han aplicado algoritmos no supervisados de agrupamiento y asociación para describir relaciones de interés en los datos.

Se pretende construir un modelo que permita predecir la categoría de las instancias en función de una serie de atributos de entrada. En el caso de WEKA, la clase es simplemente uno de los atributos simbólicos disponibles, que se convierte en la variable objetivo a predecir. Por defecto, es el último atributo (última columna) a no ser que se indique otro explícitamente. La configuración de la clasificación se efectúa con la ventana siguiente:

Selección y configuración del algoritmo de clasificación



la parte superior, como es habitual sirve para seleccionar el algoritmo de clasificación y configurarlo. El resto de elementos a definir en esta ventana se describen a continuación.

1.8.1. Modos de evaluación del clasificador

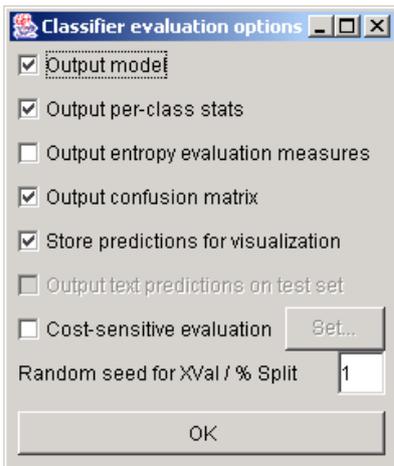
El resultado de aplicar el algoritmo de clasificación se efectúa comparando la clase predicha con la clase real de las instancias. Esta evaluación puede realizarse de diferentes modos, según la selección en el cuadro **Test options**:

- **Use training set:** esta opción evalúa el clasificador sobre el mismo conjunto sobre el que se construye el modelo predictivo para determinar el error, que en este caso se denomina "error de resustitución". Por tanto, esta opción puede proporcionar una estimación demasiado optimista del comportamiento del clasificador, al evaluarlo sobre el mismo conjunto sobre el que se hizo el modelo.
- **Supplied test set:** evaluación sobre conjunto independiente. Esta opción permite cargar un conjunto nuevo de datos. Sobre cada dato se realizará una predicción de clase para contar los errores.
- **Cross-validation:** evaluación con validación cruzada. Esta opción es la más elaborada y costosa. Se realizan tantas evaluaciones como se indica en el parámetro **Folds**. Se dividen las instancias en tantas carpetas como indica este parámetro y en cada evaluación se toman las instancias de cada carpeta como datos de test, y el resto como datos de entrenamiento para

construir el modelo. Los errores calculados son el promedio de todas las ejecuciones.

- **Percentage split** : esta opción divide los datos en dos grupos, de acuerdo con el porcentaje indicado (%). El valor indicado es el porcentaje de instancias para construir el modelo, que a continuación es evaluado sobre las que se han dejado aparte. Cuando el número de instancias es suficientemente elevado, esta opción es suficiente para estimar con precisión las prestaciones del clasificador en el dominio.

Además de estas opciones para seleccionar el modo de evaluación, el botón **More Options** abre un cuadro con otras opciones adicionales:



Output model: permite visualizar (en modo texto y, con algunos algoritmos, en modo gráfico) el modelo construido por el clasificador (árbol, reglas, etc.)

Output per-class stats: obtiene estadísticas de los errores de clasificación por cada uno de los valores que toma el atributo de clase

Output entropy evaluation measures: generaría también medidas de evaluación de entropía

Store predictions for visualization: permite analizar los errores de clasificación en una ventana de visualización

Cost-sensitive evaluation: con esta opción se puede especificar una función con costes relativos de los diferentes errores, que se rellena con el botón **Set**

en nuestro ejemplo utilizaremos los valores por defecto de estas últimas opciones.

Evaluación del clasificador en ventana de texto

Una vez se ejecuta el clasificador seleccionado sobre los datos de la relación, en la ventana de texto de la derecha aparece información de ejecución, el modelo generado con todos los datos de entrenamiento y los resultados de la

evaluación. Por ejemplo, al predecir el atributo "presentado", con un árbol de decisión de tipo J48, aparece el modelo textual siguiente:

```
J48 pruned tree
-----
cal_prueba <= 0: NO (153.0)
cal_prueba > 0: SI (18649.0/2.0)

Number of Leaves   :    2
Size of the tree   :    3
```

Se obtiene a partir de los datos esta relación trivial, salvo dos únicos casos de error: los presentados son los que tienen una calificación superior a 0. Con referencia al informe de evaluación del clasificador, podemos destacar tres elementos:

- **Resumen (Summary):** es el porcentaje global de errores cometidos en la evaluación
- **Precisión detallada por clase:** para cada uno de los valores que puede tomar el atributo de clase: el porcentaje de instancias con ese valor que son correctamente predichas (TP: true positives), y el porcentaje de instancias con otros valores que son incorrectamente predichas a ese valor aunque tenían otro (FP: false positives). Las otras columnas, *precision*, *recall*, *F-measure*, se relacionan con estas dos anteriores.
- **Matriz de confusión:** aquí aparece la información detallada de cuantas instancias de cada clase son predichas a cada uno de los valores posibles. Por tanto, es una matriz con N^2 posiciones, con N el número de valores que puede tomar la clase. En cada fila i , $i=1\dots N$, aparecen las instancias que realmente son de la clase i , mientras que las columnas j , $j=1\dots N$, son las que se han predicho al valor j de la clase. En el ejemplo anterior, la matriz de confusión que aparece es la siguiente:

```
=== Confusion Matrix ===
      a      b  <-- classified as
18647   0 |      a = SI
      2  153 |      b = NO
```

por tanto, los valores en la diagonal son los aciertos, y el resto de valores son los errores. De los 18647 alumnos presentados, todos son correctamente clasificados, mientras que de los 155 no presentados, hay 153 correctamente clasificados y 2 con error.

Lista de resultados

Al igual que con otras opciones de análisis, la ventana izquierda de la lista de resultados contiene el resumen de todas las aplicaciones de clasificadores sobre conjuntos de datos en la sesión del *Explorer*. Puede accederse a esta

lista para presentar los resultados, y al activar el botón derecho aparecen diferentes opciones de visualización, entre las que podemos destacar las siguientes:

- Salvar y cargar modelos: **Load model, Save model**. Estos modelos pueden recuperarse de fichero para posteriormente aplicarlos a nuevos conjuntos de datos
- Visualizar árbol y errores de predicción: **Visualize tree, Visualize classifier errors,...**

el árbol (permite almacenar Una vez se ejecuta el clasificador seleccionado sobre los datos de la relación,

1.8.2. Selección y configuración de clasificadores

Vamos a ilustrar la aplicación de algoritmos de clasificación a diferentes problemas de predicción de atributos definidos sobre los datos de entrada en este ejemplo. El problema de clasificación siempre se realiza sobre un atributo simbólico, en el caso de utilizar un atributo numérico se precisa por tanto discretizarlo antes en intervalos que representarán los valores de clase.

En primer lugar efectuaremos análisis de predicción de la calificación en la prueba de selectividad a partir de los siguientes atributos: año, convocatoria, localidad, opción, presentado y nota de bachillerato. Se van a realizar dos tipos de predicciones: aprobados, e intervalos de clasificación. Por tanto tenemos que aplicar en primer lugar una combinación de filtros que elimine los atributos no deseados relativos a calificaciones parciales y asignaturas opcionales, y un filtro que discretice la calificación en la prueba en dos partes:

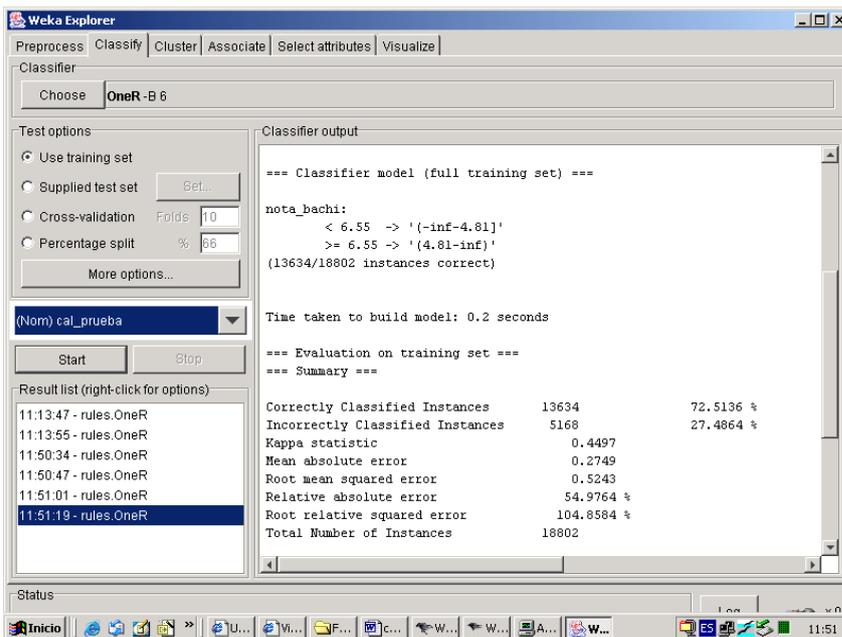
The screenshot shows the Weka Explorer interface with the 'Visualize' tab selected. The 'Current relation' is 'selectividad-weka.filters.unsupervised.attribute.Rem...' with 18802 instances and 7 attributes. The 'Selected attribute' is 'cal_prueba' (Nominal, 2 distinct values). The 'Visualize All' button is active, and a bar chart shows two categories: '[-inf-4.81]' with 9476 instances and '(4.81-inf]' with 9326 instances.

Label	Count
'[-inf-4.81]'	9476
'(4.81-inf]'	9326

obsérvese que se prefiere realizar las predicciones sobre la calificación en la prueba, puesto que la calificación final depende explícitamente de la nota del bachillerato.

Clasificador “OneR”

Este es uno de los clasificadores más sencillos y rápidos, aunque en ocasiones sus resultados son sorprendentemente buenos en comparación con algoritmos mucho más complejos. Simplemente selecciona el atributo que mejor “explica” la clase de salida. Si hay atributos numéricos, busca los umbrales para hacer reglas con mejor tasa de aciertos. Lo aplicaremos al problema de predicción de aprobados en la prueba a partir de los atributos de entrada, para llegar al resultado siguiente:



The screenshot shows the Weka Explorer interface with the OneR classifier selected. The classifier output window displays the following information:

```

=== Classifier model (full training set) ===
nota_bachi:
  < 6.55 -> '(-inf-4.81]'
  >= 6.55 -> '(4.81-inf)'
(13634/18802 instances correct)

Time taken to build model: 0.2 seconds

=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances      13634      72.5136 %
Incorrectly Classified Instances    5168      27.4864 %
Kappa statistic                    0.4497
Mean absolute error                 0.2749
Root mean squared error             0.5243
Relative absolute error             54.9764 %
Root relative squared error         104.8584 %
Total Number of Instances          18802
  
```

The result list shows several runs of the OneR classifier, with the most recent run at 11:51:19 selected.

por tanto, el algoritmo llega a la conclusión que la mejor predicción posible con un solo atributo es la nota del bachillerato, fijando el umbral que determina el éxito en la prueba en 6.55. La tasa de aciertos sobre el propio conjunto de entrenamiento es del 72.5%. Compárese este resultado con el obtenido mediante ejecución sobre instancias independientes.

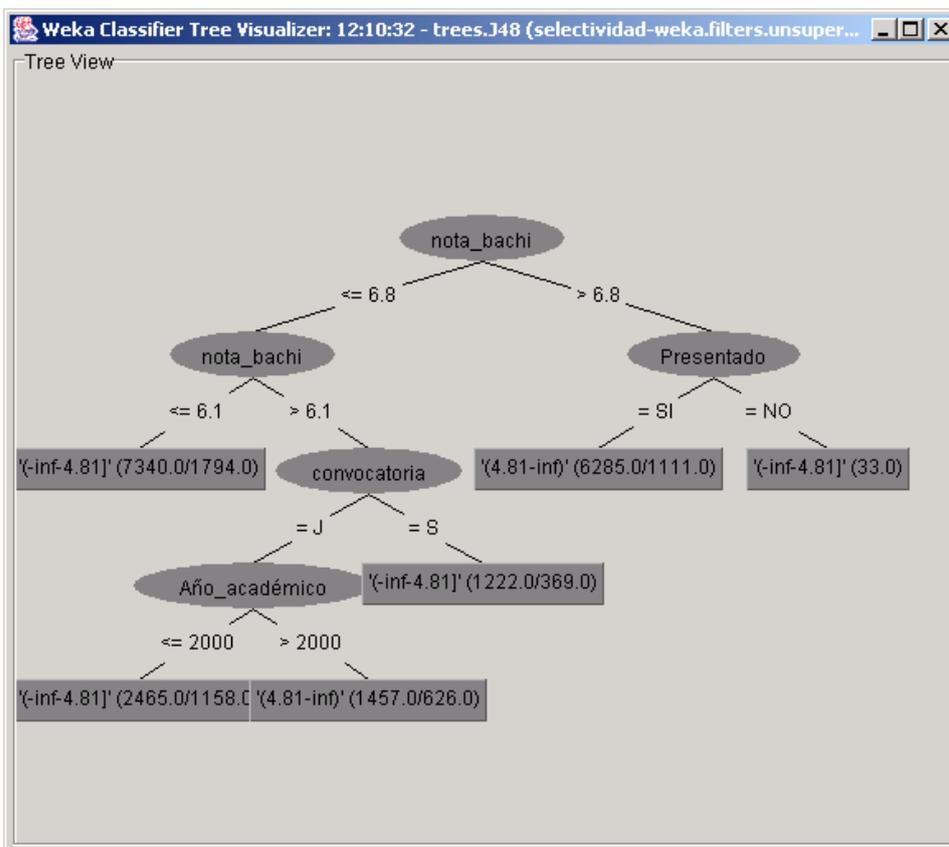
Clasificador como árbol de decisión: J48

El algoritmo J48 de WEKA es una implementación del algoritmo C4.5, uno de los algoritmos de minería de datos que más se ha utilizado en multitud de aplicaciones. No vamos a entrar en los detalles de todos los parámetros de configuración, dejándolo para el lector interesado en los detalles de este algoritmo, y únicamente resaltaremos uno de los más importantes, el factor de confianza para la poda, **confidence level**, puesto que influye notoriamente en el tamaño y capacidad de predicción del árbol construido.

Una explicación simplificada de este parámetro de construcción del árbol es la siguiente: para cada operación de poda, define la probabilidad de error que se permite a la hipótesis de que el empeoramiento debido a esta operación es significativo. Cuanto más baja se haga esa probabilidad, se exigirá que la diferencia en los errores de predicción antes y después de podar sea más

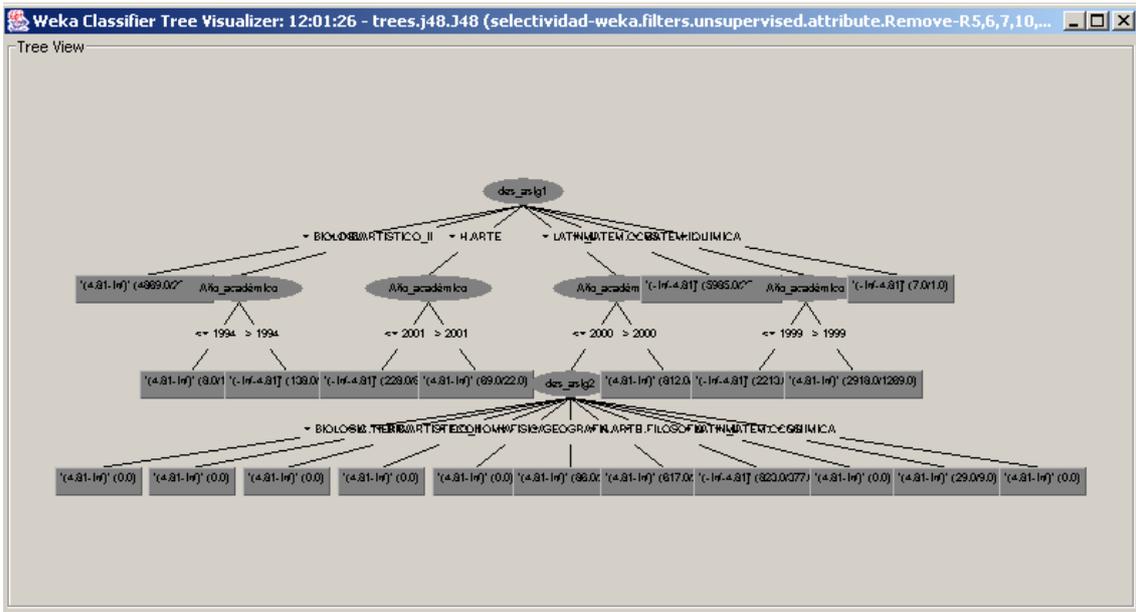
significativa para no podar. El valor por defecto de este factor es del 25%, y conforme va bajando se permiten más operaciones de poda y por tanto llegar a árboles cada vez más pequeños. Otra forma de variar el tamaño del árbol es a través de un parámetro que especifica el mínimo número de instancias por nodo, si bien es menos elegante puesto que depende del número absoluto de instancias en el conjunto de partida.

Construiremos el árbol de decisión con los parámetros por defecto del algoritmo J48: se llega a un clasificador con más de 250 nodos, con una probabilidad de acierto ligeramente superior al del clasificador *OneR*. Modifique ahora la configuración del algoritmo para llegar a un árbol más manejable, como el que se presenta a continuación



Obsérvese que este modelo es un refinamiento del generado con *OneR*, que supone una mejora moderada en las prestaciones. De nuevo los atributos más importantes son la calificación de bachillerato, la convocatoria, y después el año, antes que la localidad o las opciones. Analice las diferencias con evaluación independiente y validación cruzada, y compárelas con las del árbol más complejo con menos poda.

Podría ser de interés analizar el efecto de las opciones y asignaturas seleccionadas sobre el éxito en la prueba, para lo cual quitaremos el atributo más importante, nota de bachillerato. Llegamos a un árbol como el siguiente, en el que lo más importante es la primera asignatura optativa, en diferentes combinaciones con el año y segunda asignatura optativa:



Este resultado generado por el clasificador puede comprobarse si se analizan los histogramas de cada variable y visualizando el porcentaje de aprobados con el color, que esta variable es la que mejor separa las clases, no obstante, la precisión apenas supera el 55%.

Otros problemas de clasificación pueden formularse sobre cualquier atributo de interés, a continuación mostramos algunos ejemplos a título ilustrativo.

Clasificación multinivel de las calificaciones

el problema anterior puede intentar refinarse y dividir el atributo de interés, la calificación final, en más niveles, en este caso 5. Los resultados se muestran a continuación

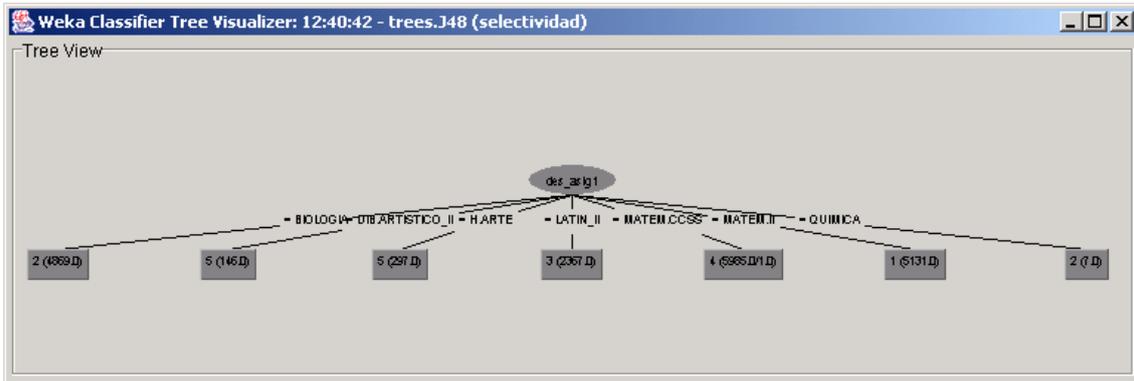
oneR

J48

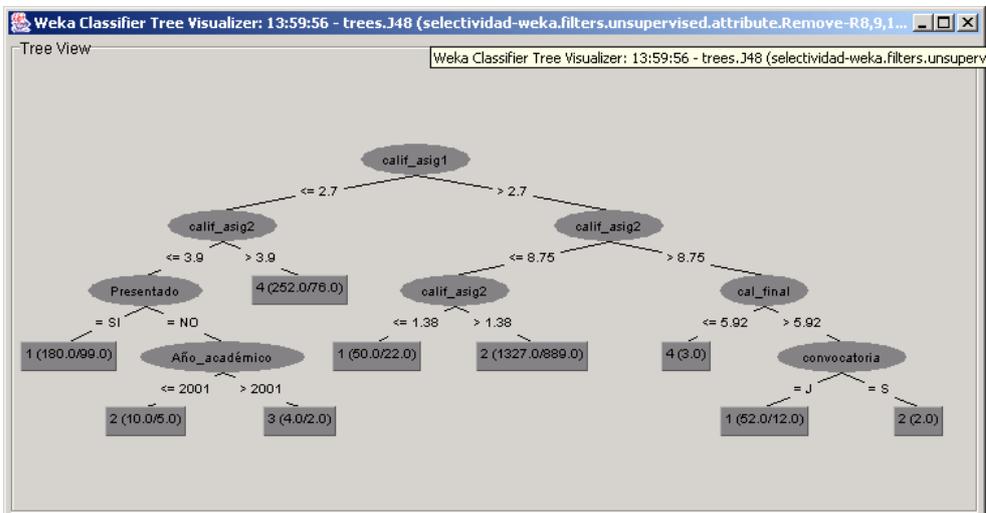
La precisión alcanzada es tan sólo del 60%, indicando que hay bastante incertidumbre una vez generada la predicción con los modelos anteriores.

Predicción de la opción

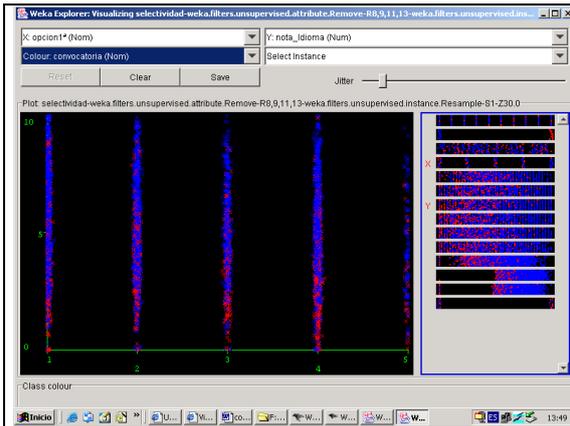
Si dejamos todos los atributos en la muestra y aplicamos el clasificador a la opción cursado, se desvela una relación trivial entre opción y asignaturas en las opciones que predice con prácticamente el 100% de los casos.



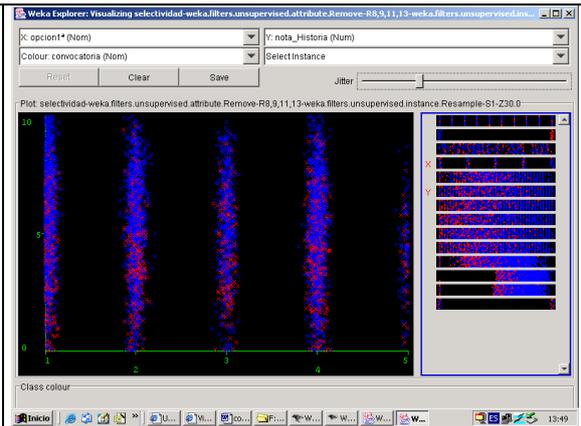
A continuación eliminamos estos designadores con un filtro de atributos. Si aplicamos el algoritmo J48 sobre los datos filtrados, llegamos a un árbol de más de 400 nodos, y con muchísimo sobre-ajuste (observe la diferencia de error de predicción sobre el conjunto de entrenamiento y sobre un conjunto independiente). Forzando la poda del árbol, llegamos al modelo siguiente:



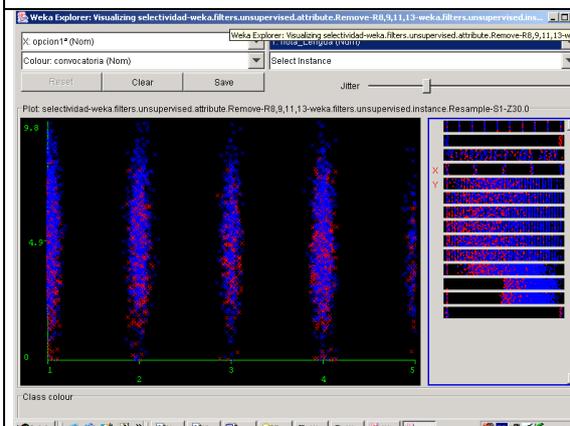
los atributos más significativos para separar las opciones son precisamente las calificaciones en las asignaturas optativas, pero apenas predice correctamente un 40% de los casos. Por tanto, vemos que no hay una relación directa entre opciones y calificaciones en la prueba, al menos relaciones que se puedan modelar con los algoritmos de clasificación disponibles. Si nos fijamos en detalle en las calificaciones en función de las opciones, podríamos determinar que apenas aparecen diferencias aparecen en los últimos percentiles, a la vista de las gráficas siguientes:



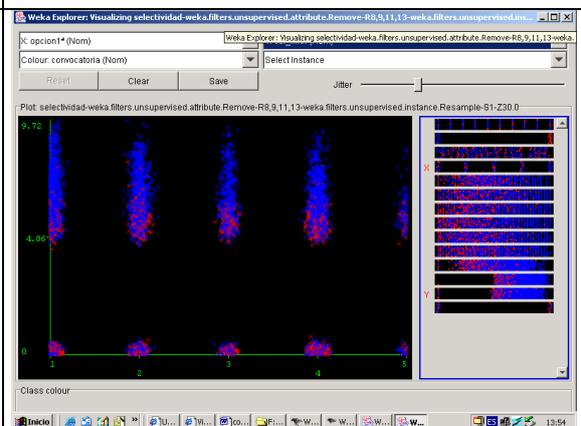
nota historia



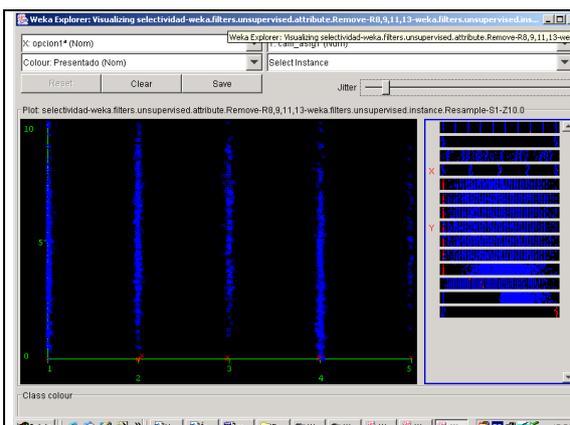
nota idioma



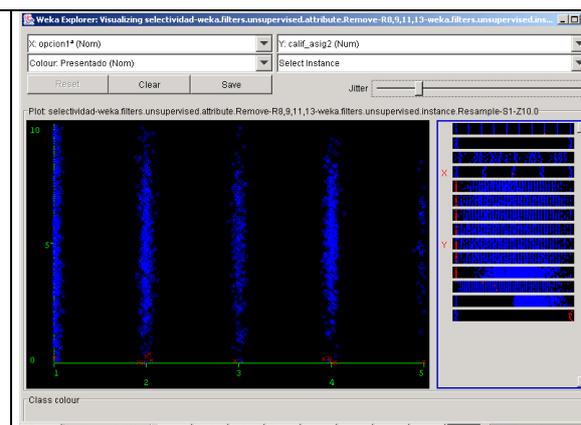
nota lengua



nota final



nota asig 1



nota asig 2

nota asig3

Vemos que las diferencias no son significativas, salvo quizá en los últimos percentiles.

Predicción de localidad y opción

La clasificación se puede realizar sobre cualquier atributo disponible. Con el número de atributos reducido a tres, localidad, opción y calificación (aprobados y suspensos), vamos a buscar modelos de clasificación, para cada uno de los atributos:

Es decir, la opción 1 y 2 aparecen mayoritariamente en Leganés, y las opciones 3 y 4 más en los alumnos que aprobaron la prueba en Leganés. No obstante, obsérvese que los errores son tan abrumadores (menos del 30% de aciertos) que cuestionan fuertemente la validez de estos modelos.

43

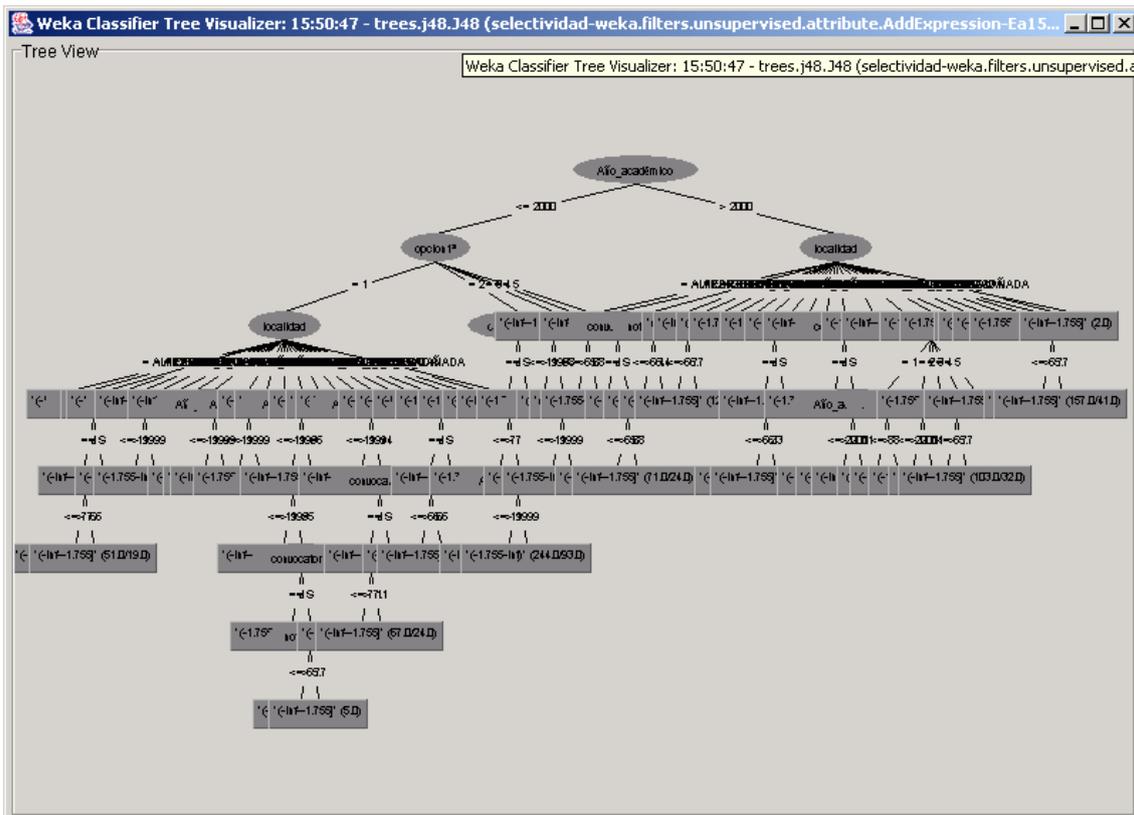
Mejora en la prueba

Un problema de clasificación interesante puede ser determinar qué alumnos tienen más "éxito" en la prueba, en el sentido de mejorar su calificación de bachillerato con la calificación en la prueba. Para ello utilizaremos el atributo "mejora", introducido en la sección 1.4.2.3, y lo discretizamos en dos valores de la misma frecuencia (obtenemos una mediana de -1.75, de manera que dividimos los alumnos en dos grupos: los que obtienen una diferencia menor a este valor y superior a este valor, para diferenciar los alumnos según el resultado se atenga más o menos a sus expectativas. Evidentemente, para evitar construir modelos triviales, tenemos que eliminar los atributos relacionados con las calificaciones en la prueba, para no llegar a la relación que acabamos de construir entre la variable calculada y las originales. Vamos a preparar el problema de clasificación con los siguientes atributos:

Atributos: 7

- Año_académico
- convocatoria
- localidad
- opcion1^a
- nota_bachi
- Presentado
- mejora

Llegamos al siguiente árbol de clasificación.



Es decir, los atributos que más determinan el "éxito" en la prueba son: año académico, opción y localidad. Para estos resultados tenemos una precisión, con evaluación sobre un conjunto independiente, en torno al 60%, por lo que sí podríamos tomarlo en consideración.

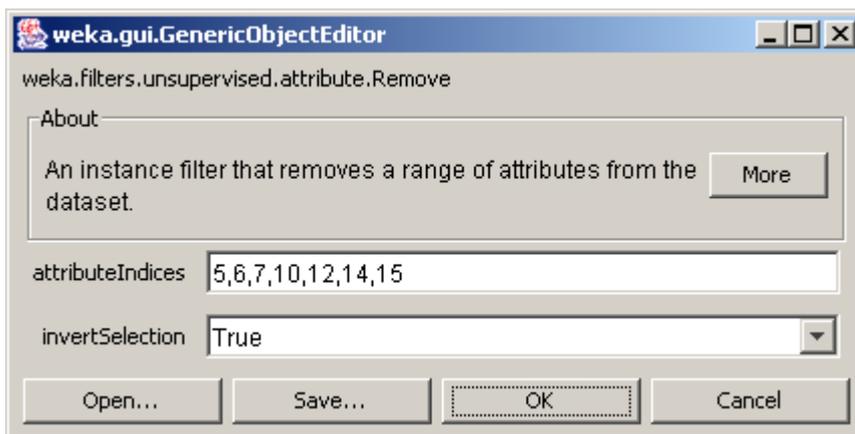
1.8.3. Predicción numérica

La predicción numérica se define en WEKA como un caso particular de clasificación, en el que la clase es un valor numérico. No obstante, los algoritmos integrados para clasificar sólo admiten clases simbólicas y los algoritmos de predicción numéricas, que aparecen mayoritariamente en el apartado *classifiers->functions*, aunque también en *classifiers->trees*.

Vamos a ilustrar algoritmos de predicción numérica en WEKA con dos tipos de problemas. Por un lado, "descubrir" relaciones deterministas que aparecen entre variables conocidas, como calificación en la prueba con respecto a las parciales y la calificación final con respecto a la prueba y bachillerato, y buscar otros modelos de mayor posible interés.

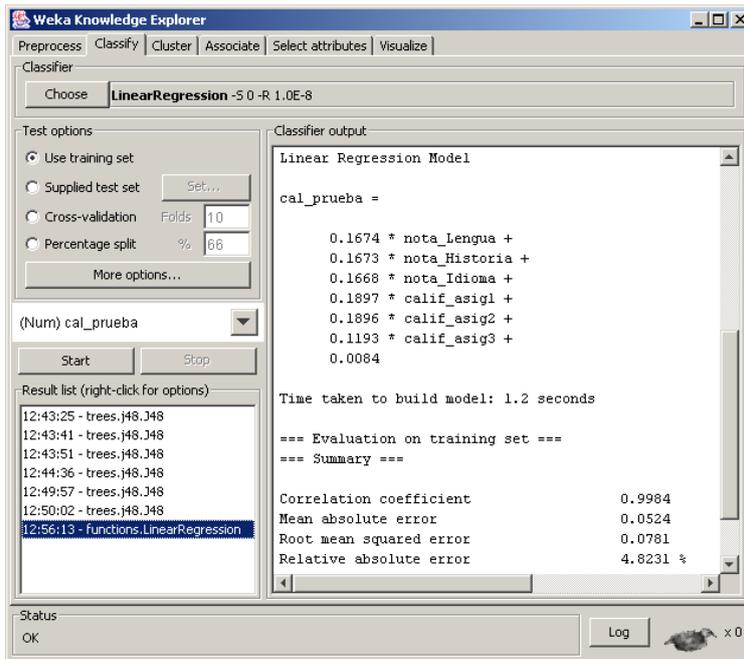
Relación entre calificación final y parciales

Seleccionamos los atributos con las 6 calificaciones parciales y la calificación en la prueba:



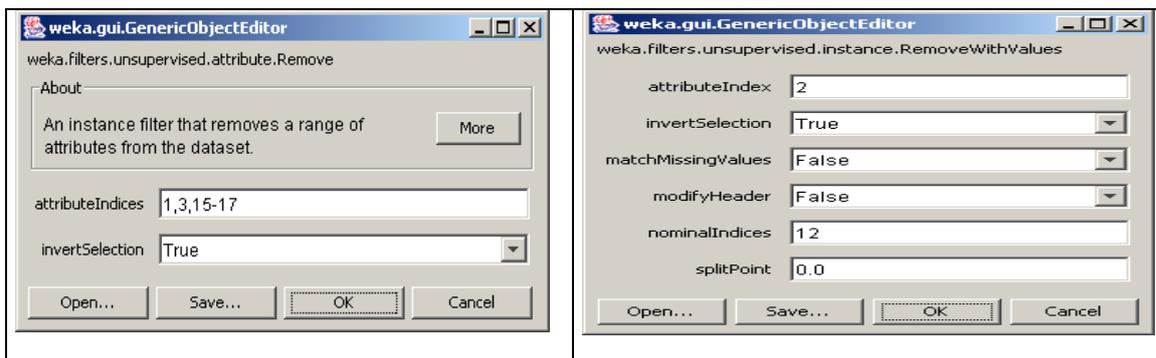
Vamos a aplicar el modelo de predicción más popular: regresión simple, que construye un modelo lineal del atributo clase a partir de los atributos de entrada: **functions->LinearRegression**

Como resultado, aparece la relación con los pesos relativos de las pruebas parciales sobre la calificación de la prueba:

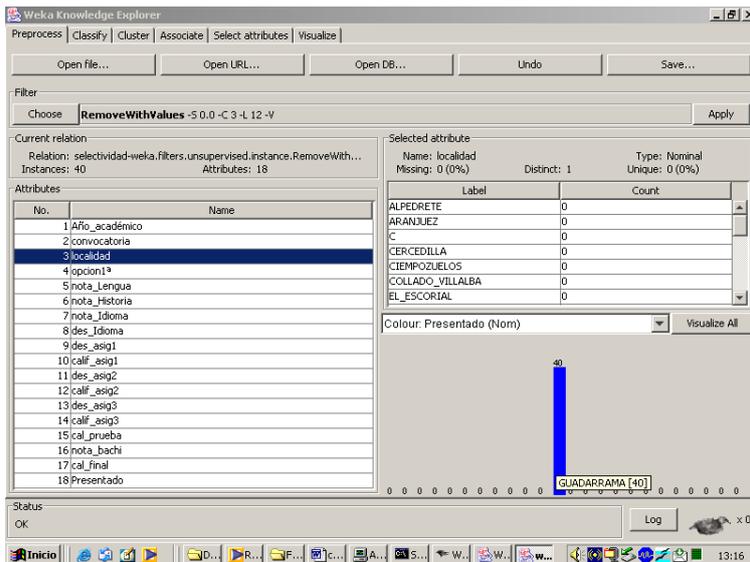


Hay que observar que en los problemas de predicción la evaluación cambia, apareciendo ahora el coeficiente de correlación y los errores medio y medio cuadrático, en términos absolutos y relativos. En este caso el coeficiente de correlación es de 0.998, lo que indica que la relación es de una precisión muy notable.

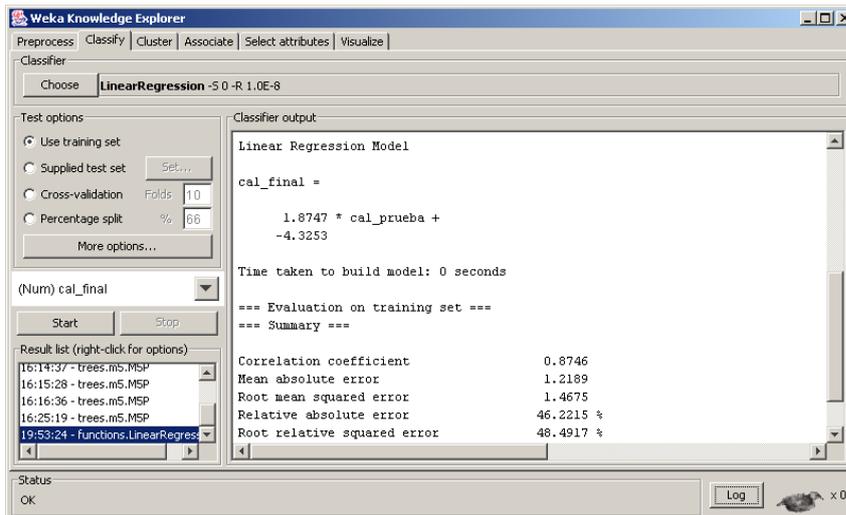
Si aplicamos ahora esta función a la relación entre calificación final con calificación en la prueba y nota de bachillerato (filtro que selecciona únicamente los atributos 15-17), podemos determinar la relación entre estas variables: qué peso se lleva la calificación de bachillerato y de la prueba en la nota final. Vamos a hacerlo primero con los alumnos de una población pequeña, de Guadarrama (posición 12 del atributo localidad). Aplicamos los filtros correspondientes para tener únicamente estos alumnos, y los atributos de calificaciones de la prueba, bachillerato y final:



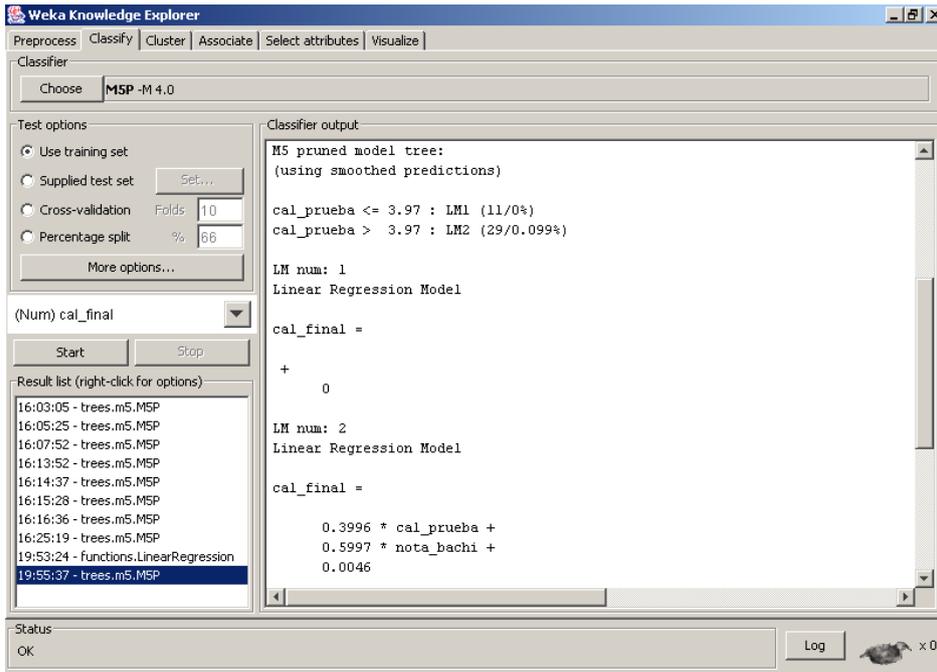
llegamos a 40 instancias:



si aplicáramos regresión lineal como en el ejemplo anterior, obtenemos el siguiente resultado:

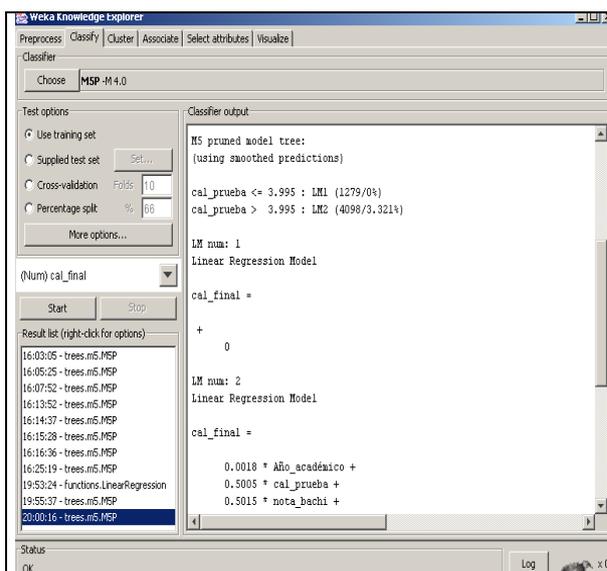


el resultado deja bastante que desear porque la relación no es lineal. Para solventarlo podemos aplicar el algoritmo M5P, seleccionado en WEKA como **trees->m5->M5P**, que lleva a cabo una regresión por tramos, con cada tramo determinado a partir de un árbol de regresión. Llegamos al siguiente resultado:

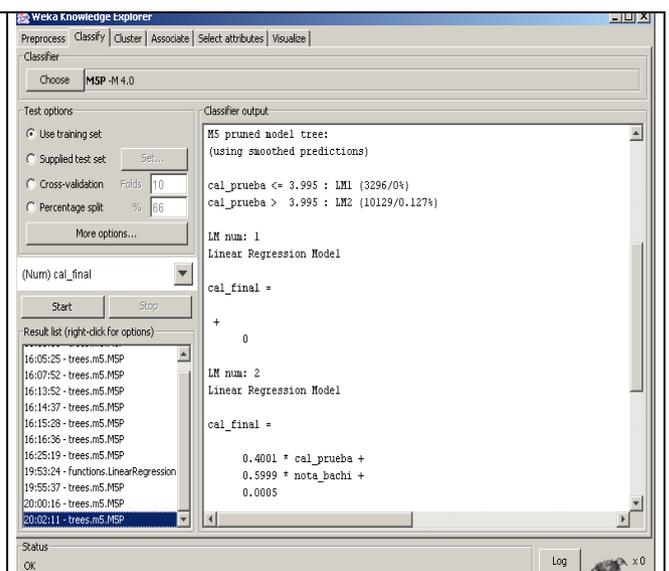


que es prácticamente la relación exacta utilizada en la actualidad: 60% nota de bachillerato y 40% de la prueba, siempre que se supere en ésta un valor mínimo de 4 puntos.

Si aplicamos este algoritmo a otros centros no siempre obtenemos este resultado, por una razón: hasta 1998 se ponderaba al 50%, y a partir de 1999 se comenzó con la ponderación anterior. Verifíquese aplicando este algoritmo sobre datos filtrados que contengan alumnos de antes de 1998 y de 1999 en adelante. En este caso, el algoritmo M5P no tiene capacidad para construir el modelo correcto, debido a la ligera diferencia en los resultados al cambiar la forma de ponderación. Los árboles obtenidos en ambos casos se incluyen a continuación:



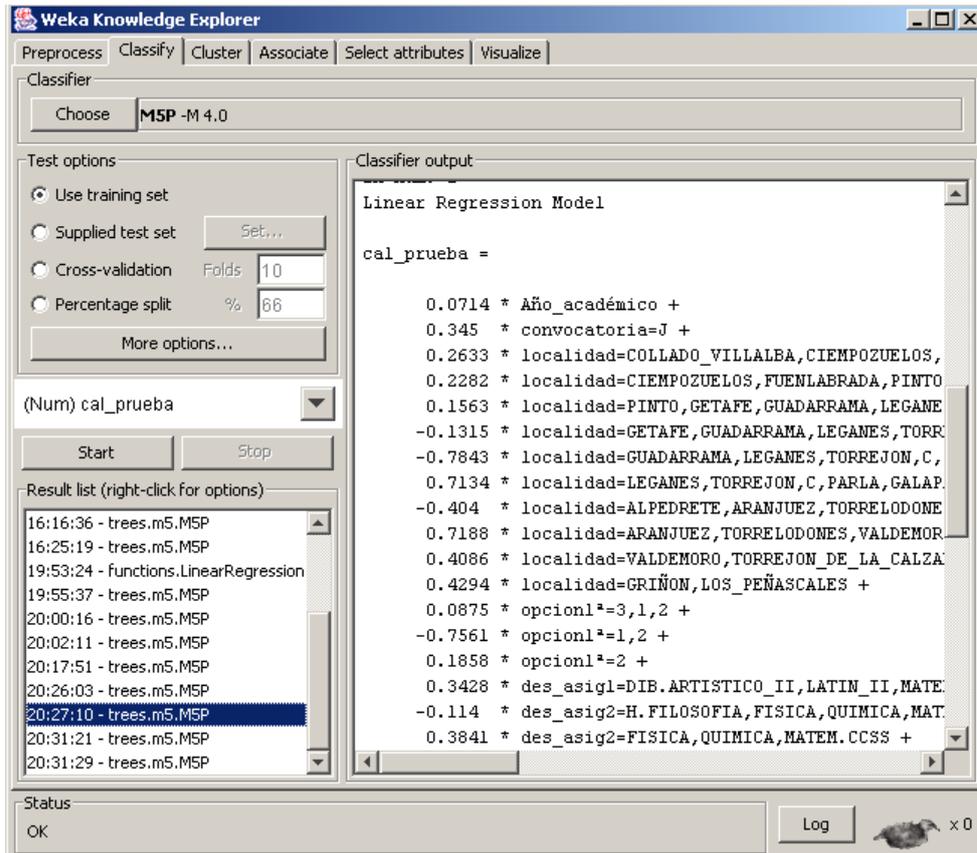
hasta 1998



de 1999 en adelante

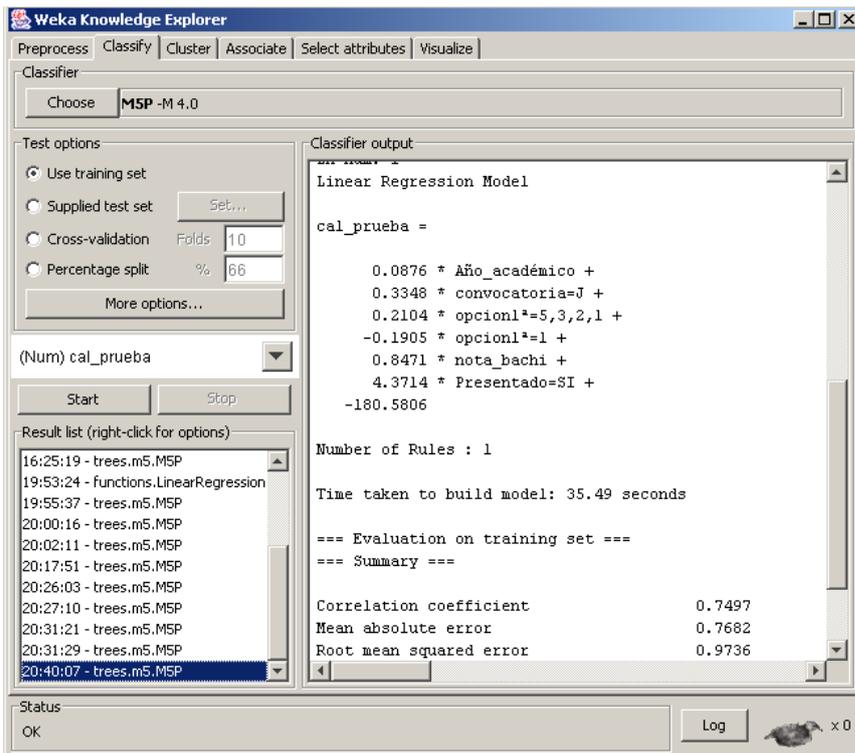
Predicción de la calificación

Vamos a aplicar ahora este modelo para intentar construir un modelo aplicación más interesante, o, al menos, analizar tendencias de interés. Se trata de intentar predecir la calificación final a partir de los atributos de entrada, los mismos que utilizamos para el problema de clasificar los alumnos que aprueban la prueba. Si aplicamos el algoritmo sobre el conjunto completo llegamos al siguiente modelo:

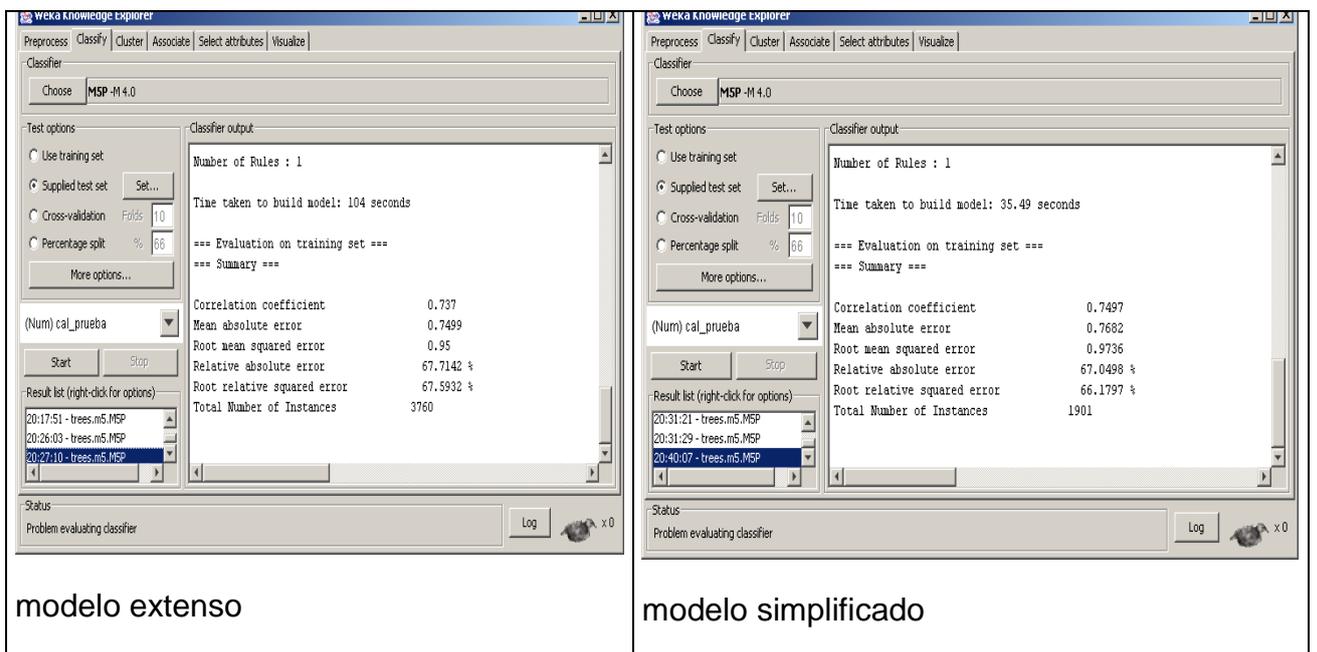


obsérvese cómo trata el algoritmo los atributos nominales para incluirlos en la regresión: ordena los valores según el valor de la magnitud a predecir (en el caso de localidad, desde Collado hasta Los Peñascales y en el de opción, ordenadas como 4^o, 5^o, 3^o, 2^o, 1^o), y va tomando variables binarias resultado de dividir en diferentes puntos, determinando su peso en la función. En esta función lo que más pesa es la convocatoria, después la nota de bachillerato, y después entran en juego la localidad, asignaturas optativas, y opción, con un modelo muy complejo.

Si simplificamos el conjunto de atributos de entrada, y nos quedamos únicamente con el año, opción, nota de bachillerato, y convocatoria, llegamos a:



este modelo es mucho más manejable. Compare los errores de predicción con ambos casos:



Correlación entre nota de bachillerato y calificación en prueba

Finalmente, es interesante a veces hacer un modelo únicamente entre dos variables para ver el grado de correlación entre ambas. Continuando con nuestro interés por las relaciones entre calificación en prueba y calificación en bachillerato, vamos a ver las diferencias por opción. Para ello filtraremos por un lado los alumnos de opción 1 y los de opción 4. A continuación dejamos

únicamente los atributos calificación en prueba y nota de bachillerato, para analizar la correlación de los modelos para cada caso.

The image shows two side-by-side screenshots of the Weka Knowledge Explorer interface. Both windows are titled 'Weka Knowledge Explorer' and show the 'Classify' tab with 'LinearRegression - S 0 - R 1.0E-8' selected as the classifier. The 'Test options' are set to 'Percentage split' at 66%.

Left Screenshot (alumnos opción 1º):

- Classifier output: Linear Regression Model
- Equation: $cal_prueba = 0.922 * nota_bach1 + -1.3478$
- Time taken to build model: 0.08 seconds
- Correlation coefficient: 0.6608
- Mean absolute error: 0.8759
- Root mean squared error: 1.1140
- Relative absolute error: 73.8933 %
- Root relative squared error: 75.0895 %
- Total Number of Instances: 1745

Right Screenshot (alumnos opción 4º):

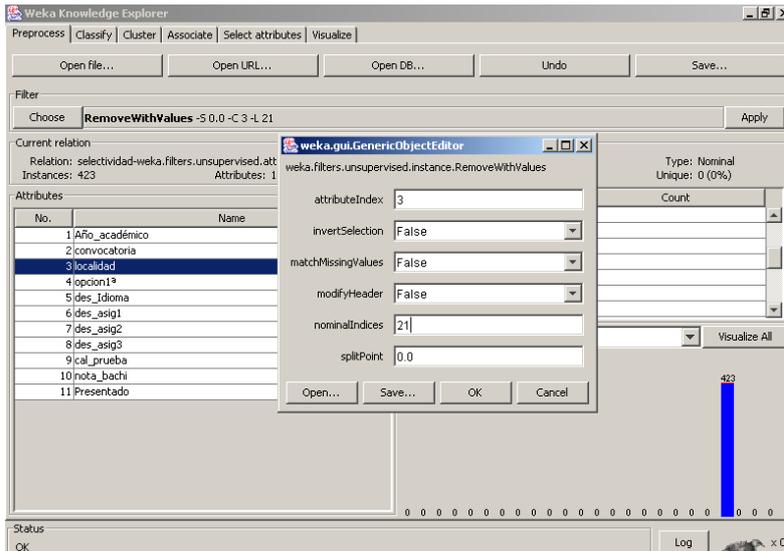
- Classifier output: Linear Regression Model
- Equation: $cal_prueba = 0.8424 * nota_bach1 + -0.842$
- Time taken to build model: 0.58 seconds
- Correlation coefficient: 0.588
- Mean absolute error: 0.7889
- Root mean squared error: 1.0441
- Relative absolute error: 79.3249 %
- Root relative squared error: 80.8973 %
- Total Number of Instances: 2035

podemos concluir que para estas dos opciones el grado de relación entre las variables sí es significativamente diferente, los alumnos que cursan la opción 1º tienen una relación más "lineal" entre ambas calificaciones que los procedentes de la opción 4º

1.8.4. Aprendizaje del modelo y aplicación a nuevos datos.

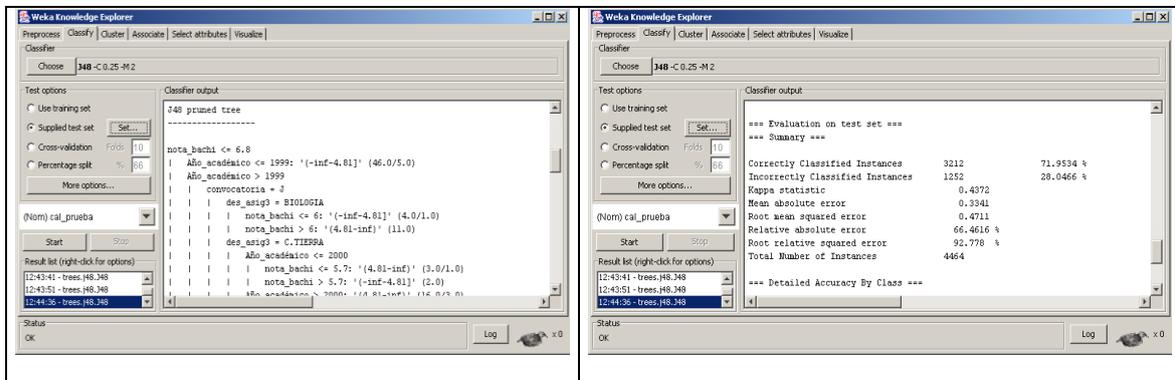
Para finalizar esta sección de clasificación, ilustramos aquí las posibilidades de construir y evaluar un clasificador de forma cruzada con dos ficheros de datos. Seleccionaremos el conjunto atributos siguiente: Año_académico, convocatoria, localidad, opcion1ª, des_Idioma, des_asig1, des_asig2, des_asig3, cal_prueba, nota_bachi, Presentado. El atributo con la calificación, "cal_prueba", lo discretizamos en dos intervalos.

Vamos a generar, con el filtro de instancias dos conjuntos de datos correspondientes a los alumnos de Getafe y Torreldones. Para ello primero seleccionamos las instancias con el atributo localidad con valor 10, lo salvamos ("datosGetafe") y a continuación las instancias con dicho atributo con valor 21 ("datosTorreldones").

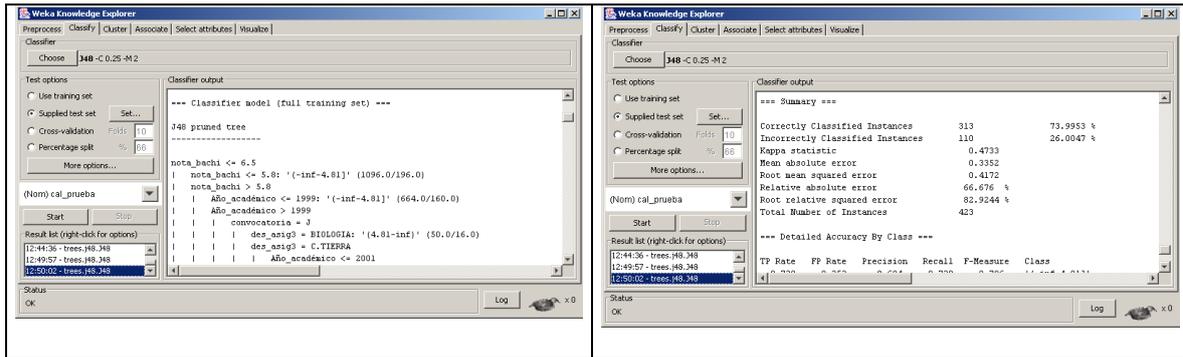


Ahora vamos a generar los modelos de clasificación de alumnos con buen y mal resultado en la prueba en la localidad de Torrelodones, para evaluarlo con los alumnos de Getafe.

Para ello en primer lugar cargamos el fichero con los alumnos de Torrelodones que acabamos de generar, “datosTorrelodones”, y lo evaluamos sobre el conjunto con alumnos de Getafe. Para ello, seleccionaremos la opción de evaluación con un fichero de datos independiente, **Supplied test set**, y fijamos con el botón **Set**, que el fichero de test es “datosGetafe”. Obsérvese el modelo generado y los resultados:



Si ahora hacemos la operación inversa, entrenar con los datos de Getafe y evaluar con los de Torrelodones, llegamos a:



Hay ligeras diferencias en los modelos generados para ambos conjuntos de datos (para los alumnos de Torrelodones, lo más importante es tener una calificación de bachillerato superior a 6.8, mientras que a los de Getafe les basta con un 6.5), y los resultados de evaluación con los datos cruzados muestran una variación muy pequeña. El modelo construido a partir de los datos de Torrelodones predice ligeramente peor los resultados de Getafe que a la inversa.

1.9. Selección de atributos

Esta última sección permite automatizar la búsqueda de subconjuntos de atributos más apropiados para "explicar" un atributo objetivo, en un sentido de clasificación supervisada: permite explorar qué subconjuntos de atributos son los que mejor pueden clasificar la clase de la instancia. Esta selección "supervisada" aparece en contraposición a los filtros de preprocesado comentados en la sección 1.4.2, que se realizan de forma independiente al proceso posterior, razón por la que se etiquetaron como "no supervisados".

La selección supervisada de atributos tiene dos componentes:

- Método de Evaluación (**Attribute Evaluator**): es la función que determina la calidad del conjunto de atributos para discriminar la clase.
- Método de Búsqueda (**Search Method**): es la forma de realizar la búsqueda de conjuntos. Como la evaluación exhaustiva de todos los subconjuntos es un problema combinatorio inabordable en cuanto crece el número de atributos, aparecen estrategias que permiten realizar la búsqueda de forma eficiente

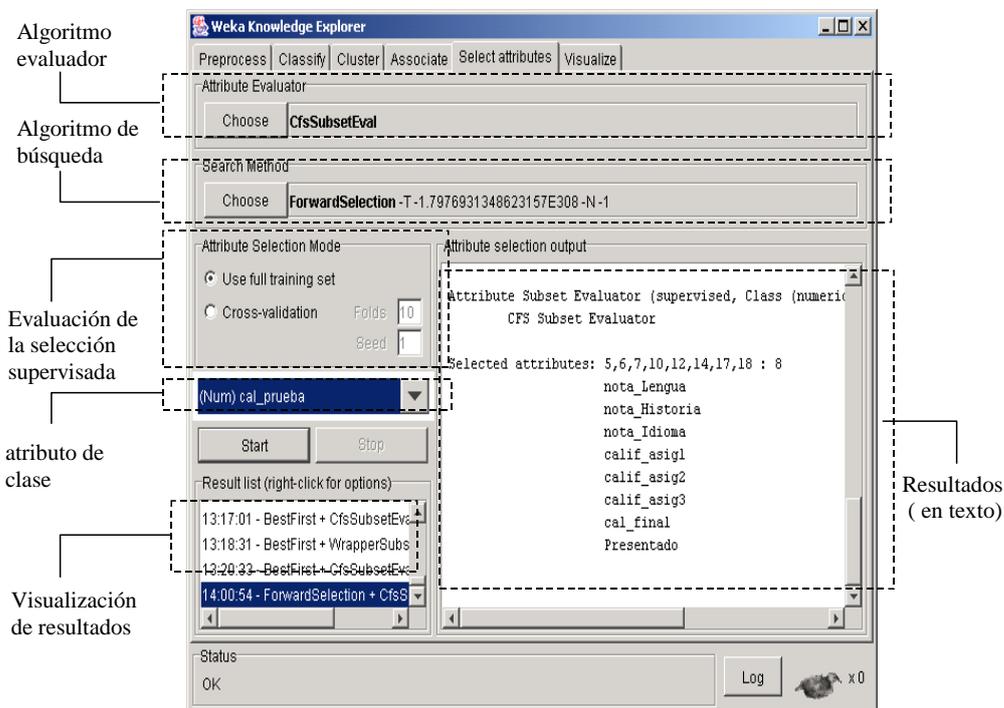
De los métodos de evaluación, podemos distinguir dos tipos: los métodos que directamente utilizan un clasificador específico para medir la calidad del subconjunto de atributos a través de la tasa de error del clasificador, y los que no. Los primeros, denominados métodos "wrapper", porque "envuelven" al clasificador para explorar la mejor selección de atributos que optimiza sus prestaciones, son muy costosos porque necesitan un proceso completo de entrenamiento y evaluación en cada paso de búsqueda. Entre los segundos podemos destacar el método "**CfsSubsetEval**", que calcula la correlación de la

clase con cada atributo, y eliminan atributos que tienen una correlación muy alta como atributos redundantes.

En cuanto el método de búsqueda, vamos a mencionar por su rapidez el "**ForwardSelection**", que es un método de búsqueda subóptima en escalada, donde elige primero el mejor atributo, después añade el siguiente atributo que más aporta y continua así hasta llegar a la situación en la que añadir un nuevo atributo empeora la situación. Otro método a destacar sería el "**BestSearch**", que permite buscar interacciones entre atributos más complejas que el análisis incremental anterior. Este método va analizando lo que mejora y empeora un grupo de atributos al añadir elementos, con la posibilidad de hacer retrocesos para explorar con más detalle. El método "**ExhaustiveSearch**" simplemente enumera todas las posibilidades y las evalúa para seleccionar la mejor

Por otro lado, en la configuración del problema debemos seleccionar qué atributo objetivo se utiliza para la selección supervisada, en la ventana de selección, y determinar si la evaluación se realizará con todas las instancias disponibles, o mediante validación cruzada.

Los elementos por tanto a configurar en esta sección se resumen en la figura siguiente:



Siguiendo con nuestro ejemplo, vamos a aplicar búsqueda de atributos para "explicar" algunos atributos objetivo. Para obtener resultados sin necesidad de mucho tiempo, vamos a seleccionar los algoritmos más eficientes de evaluación y búsqueda, **CfsSubsetEval** y **ForwardSelection**

Por ejemplo, para la calificación final tenemos 8 atributos seleccionados:

```
Selected attributes: 5,6,7,10,12,14,17,18 : 8
    nota_Lengua
    nota_Historia
    nota_Idioma
    calif_asig1
    calif_asig2
    calif_asig3
    cal_final
    Presentado
```

y para la opción 1 atributo:

```
Selected attributes: 9 : 1
    des_asig1
```

Por tanto, hemos llegado a los atributos que mejor explican ambos (la calificación en la prueba depende directamente de las parciales, y la opción se explica con la 1ª asignatura), si bien son relaciones bastante triviales. A continuación preparamos los datos para buscar relaciones no conocidas, quitando los atributos referentes a cada prueba parcial. Dejando como atributos de la relación:

```
Attributes: 7
    Año_académico
    convocatoria
    localidad
    opcion1ª
    cal_prueba
    nota_bachi
    Presentado
```

para la calificación final llegamos a 2 atributos:

```
Selected attributes: 6,7 : 2
    nota_bachi
    Presentado
```

y para la opción 2:

```
Selected attributes: 3,5,6 : 3
    localidad
    cal_prueba
    nota_bachi
```

No obstante, si observamos la figura de mérito con ambos problemas, que aparece en la ventana textual de resultados, vemos que este segundo es mucho menos fiable, como ya hemos comprobado en secciones anteriores.