

Práctica 1 de MINERIA DE DATOS 09/10
Preprocesamiento mediante la herramienta WEKA
Duración: 3 Sesiones
Fecha límite de entrega: 9 de Noviembre

José A. Gámez, M. Julia Flores & Pablo Bermejo

14/10/2009

1. Breve introducción a WEKA

En esta primera práctica lo primero que debéis hacer es familiarizaros con la herramienta Weka, principalmente en el manejo de su interfaz de usuario, aunque también algunas nociones sobre las facilidades que proporciona como librería de programación. Por tanto además de evaluarse el trabajo que se desarrolle, tened en cuenta que los conocimientos y habilidades adquiridas en esta práctica número uno serán muy necesarias en las prácticas futuras.

Weka es una herramienta de libre distribución realizada por la universidad de Waikato en Nueva Zelanda que se puede descargar de
<http://www.cs.waikato.ac.nz/ml/weka/>

En la distribución de este software se encuentra un manual de la interfaz Explorer que es la que utilizaremos (ExplorerGuide.pdf) y también un manual para utilizar Weka por línea de comandos (Tutorial.pdf).

Además, os proporcionamos un enlace con un útil tutorial realizado por profesorado de la Universidad Politécnica de Valencia:

<http://www.dsic.upv.es/~cferri/weka/CursDoctorat-weka.pdf>

De cualquier modo, en los siguientes subapartados os comentamos las capacidades más importantes incluyendo capturas de pantalla.

1.1. Interfaz Explorer

Al ejecutar Weka nos pararecerá la ventana que se muestra en la figura 1.1. En ella seleccionaremos el botón **Explorer** y nos aparecerá una nueva ventana que se muestra en la figura 2.

Nosotros sólo vamos a hacer uso de la interfaz gráfica del módulo Explorer, a pesar de que Weka cuenta también con una interfaz por línea de comandos y otros módulos de interfaz gráfica. Con Explorer tenemos al alcance de la mano prácticamente todas la herramientas que se pueden llegar a necesitar en el desarrollo de una tarea de minería de datos. Inicialmente la mayoría de



Figura 1: Interfaz de Weka

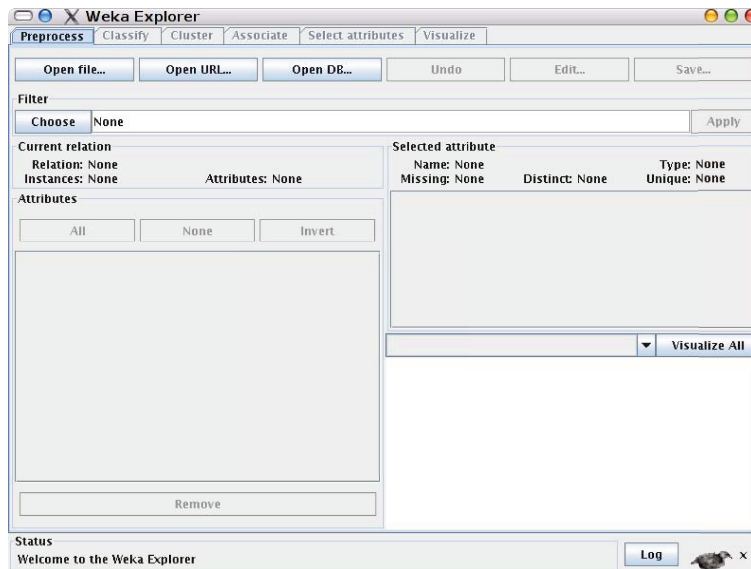


Figura 2: Interfaz Explorer

los botones y controles están deshabilitados ya que es necesario cargar una base de datos con la que trabajar.

En este ejemplo vamos utilizar una base de datos conocida como *weather*. Al cargarla podremos ver diversa información sobre ella (ver figura 3) como por ejemplo el número de instancias que la forman, los atributos así una descripción de ellos. En este caso podemos ver que la quinta variable, que será para nosotros la variable clase¹, es discreta y tiene dos estados. También podríamos ver que del resto de atributos dos de ellos son nominales o los otros numéricos.

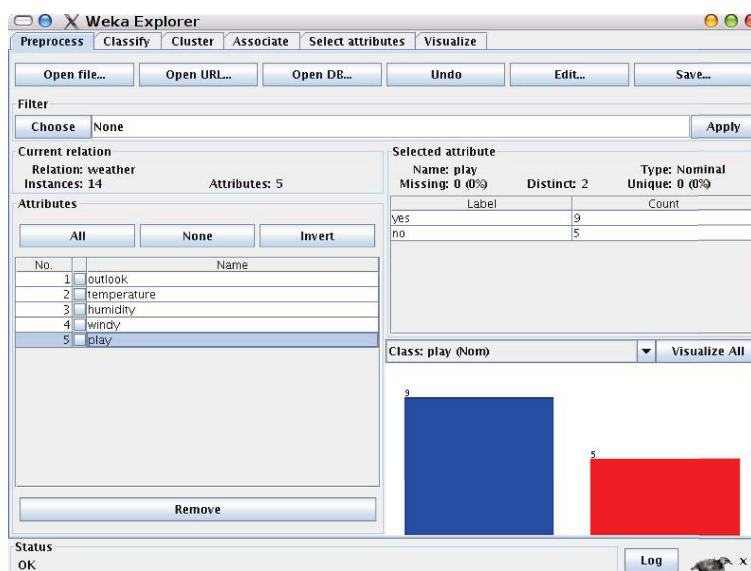


Figura 3: Base de datos weather cargada

¹Normalmente, en un escenario de clasificación o regresión se considera la variable clase como la última de la base de datos por defecto pero podemos indicar otra variable para que actúe como clase.

En esta primera pestaña de Weka también podemos realizar algunas operaciones de preprocesamiento sobre los datos. En la figura 4 se muestra una lista parcial de los filtros que se pueden elegir para procesar los datos

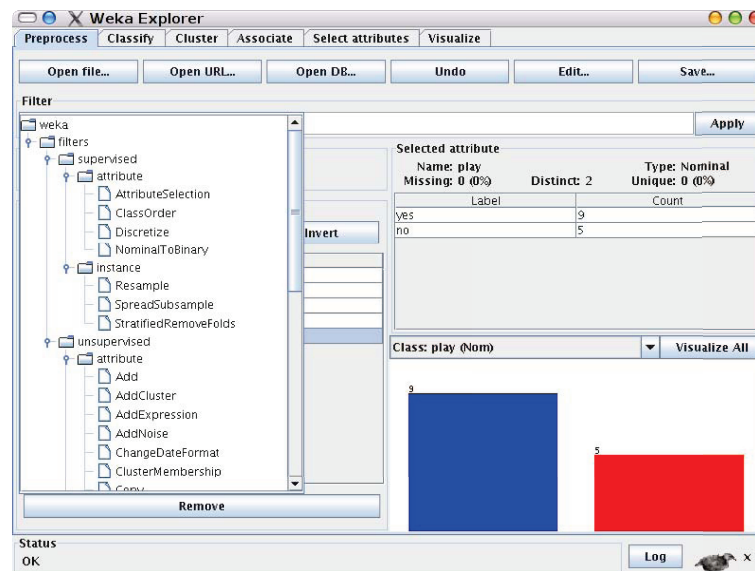


Figura 4: Lista de filtros disponibles para el preprocesamiento

Estos filtros están divididos primero entre supervisados y no supervisados. Los supervisados son aquellos algoritmos que tienen en cuenta que existe una variable clase y la utilizan para realizar un procesamiento más orientado. A su vez, dentro de esta separación tenemos otro nivel que divide los algoritmos entre los que aplican sobre atributos o sobre instancias.

Entre las opciones disponibles para atributos podemos encontrar discretización, construcción de nuevos atributos por composición, entre otros. De los aplicados a instancias tenemos por ejemplo tratamiento de las instancias con valores perdidos, codificar los datos de forma dispersa para ahorrar espacio cuando hay muchos valores a cero, etc.

Como la base de datos cargada contiene algunos atributos numéricos y dado que algunos de los algoritmos no pueden tratar este tipo de datos, vamos a optar por discretizar estos atributos. Podemos elegir el filtro de discretización no supervisado para particionar las variables en dos estados manteniendo el resto de parámetros por defecto. Como resultado dividirá el espacio continuo en dos intervalos como se muestra en la figura 5.

En la ventana principal de Weka, la siguiente pestaña está etiquetada como **Classify**. Evidentemente en este apartado podemos aplicar diversos algoritmos de clasificación sobre la base de datos. También están separados por grupos entre clasificadores bayesianos (Naive Bayes, TAN,...), funciones lineales y no lineales (redes neuronales, máquinas de soporte vectorial,...), de construcción perezosa (vecinos más cercanos,...), metaclasificadores (boosting, bagging,...), árboles (ID3, C4.5,...) y reglas (OneR, PART,...).

El siguiente apartado está dedicado al clustering. Aquí tenemos el clásico algoritmo EM junto con otros como el Cobweb o el KMeans.

Al igual que con los algoritmos de clasificación Weka nos da varias formas de validar los resultados. Se puede elegir validación contra la propia base de aprendizaje, especificar un fichero de test o utilizar validación cruzada indicando el número de particiones.

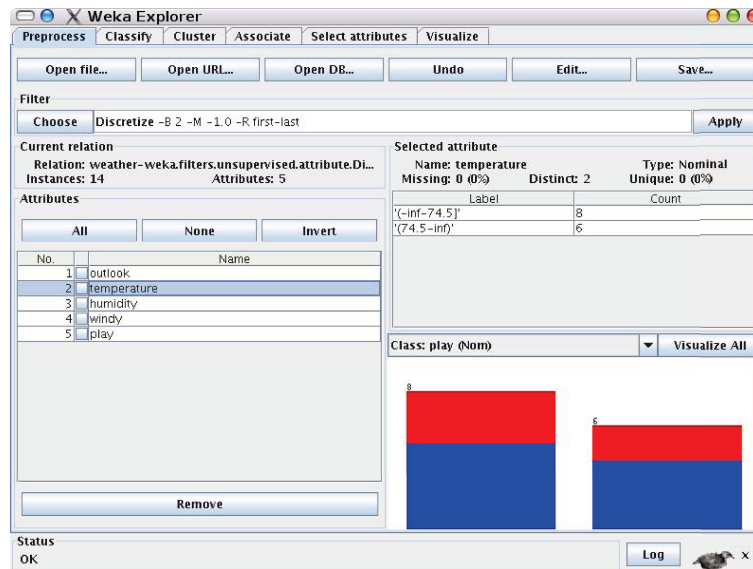


Figura 5: Aplicado un filtro de discretización

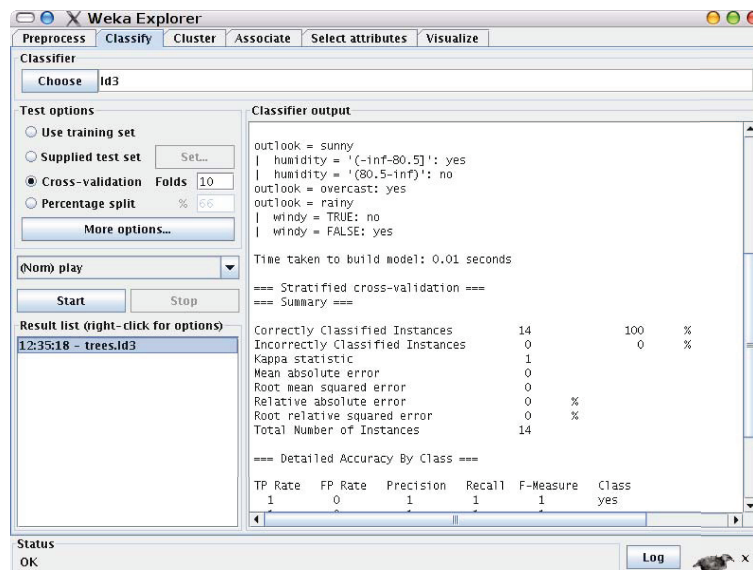


Figura 6: Apartado de clasificación

Otra forma de analizar los datos puede ser la utilización de reglas de asociación (pestaña **Associate**) para encontrar las relaciones que se esconden entre la maraña de datos. Los controles que podemos utilizar se muestran en la figura 8. En este caso contamos con algoritmos como *Apriori* y *Tertius*.

Como un complemento al apartado de preprocesamiento tenemos el de selección de variables (figura 9). Podemos utilizar estos algoritmos para intentar decidir si hay variables en nuestra base de datos que son irrelevantes o que la información que proporcionan no justifica la complejidad de considerar el atributo. Tenemos varias formas de valorar la relevancia de cada variable así como distintos algoritmos de búsqueda para encontrar el mejor subconjunto de variables en función de esta relevancia.

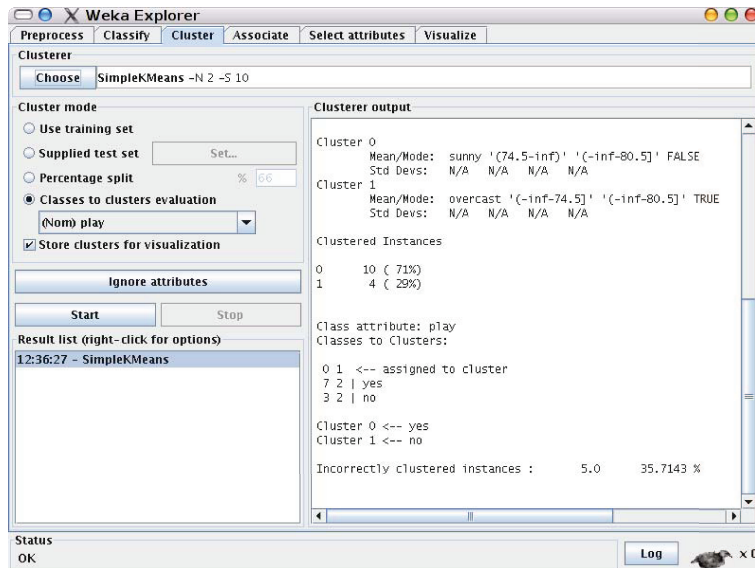


Figura 7: Apartado de clustering

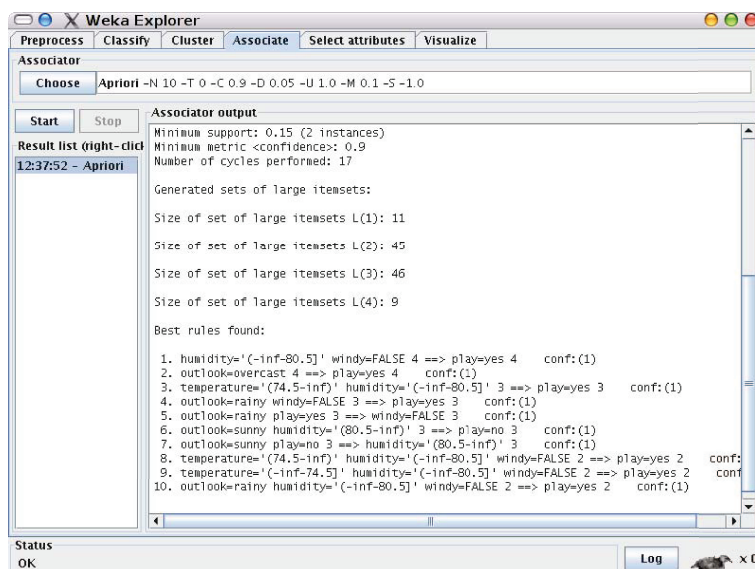


Figura 8: Apartado de reglas de asociación

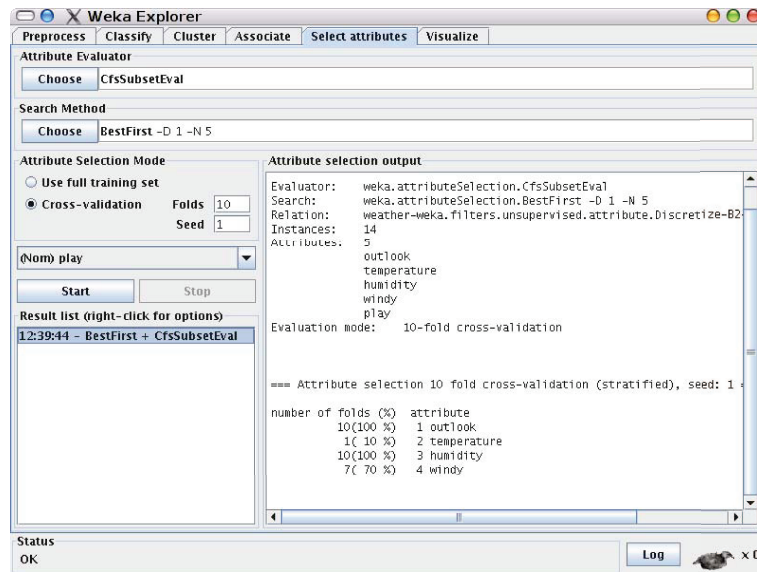


Figura 9: Apartado de selección de atributos

El último apartado es el visualización, y su razón de ser es la de poder apreciar visualmente la relación entre atributos ya que nos muestra la distribución de las instancias de la base de datos en función de los valores de cada par de variables. Inicialmente como se puede ver en la figura 10 tenemos todas las combinaciones de pares de atributos y seleccionando alguna de ellas podemos verla con más detalle como se muestra en la figura 11

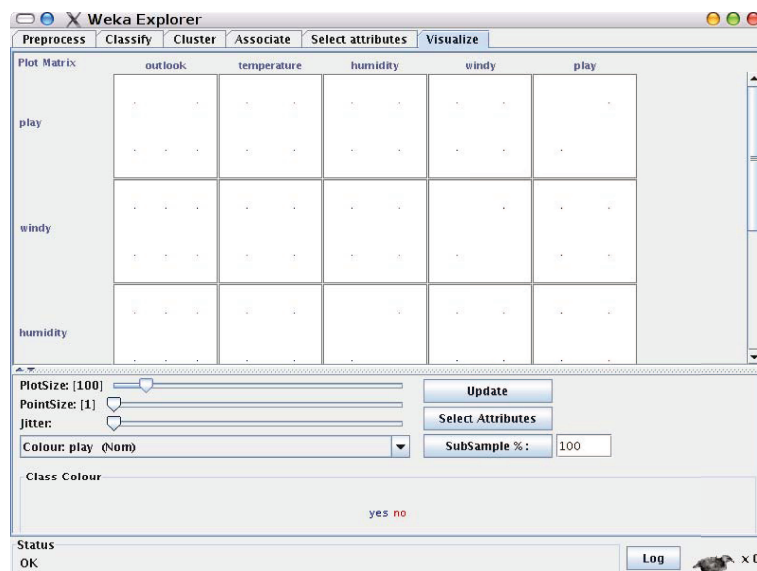


Figura 10: Apartado de visualización

1.2. Programando con Weka

Weka está constituida por una serie de clases y interfaces en Java. De ellos los más interesantes desde el punto de vista del programador son los siguientes:

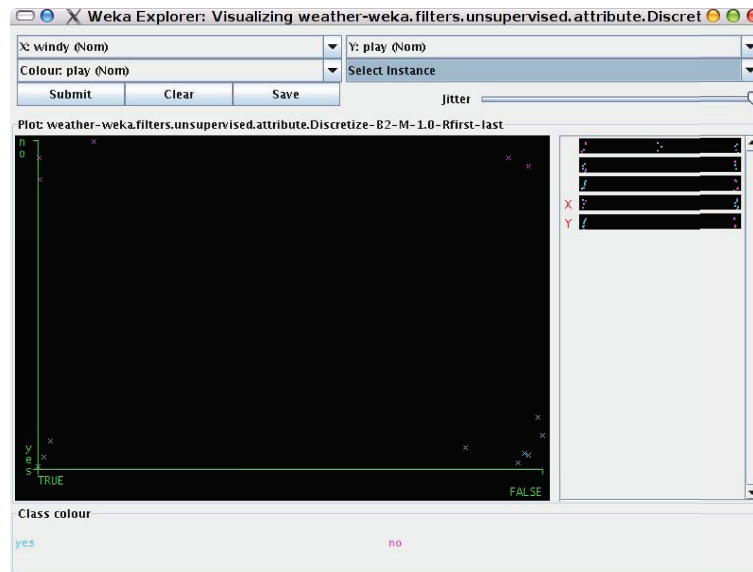


Figura 11: Visualización de la relación entre las variables *windy* y *play*

- *weka.core.Instances* Es probablemente la clase central de la arquitectura de weka ya que codifica a la base de datos con la que trabajan los algoritmos.
- *weka.core.Instance* Codifica cada fila de la base de datos.
- *weka.core.Attribute* Representa cada atributo o columna.
- *weka.core.Utils* Contiene algunas funciones de interés.

En los distintos paquetes Weka organiza las interfaces y clases abstractas que definen la funcionalidad básica de los algoritmos que se pueden implementar. Por ejemplo la clase abstracta *weka.classifiers.Classifier* es la que se ha de utilizar como base para implementar cualquier nuevo clasificador.

Weka tiene una documentación muy buena en Javadoc sobre la API que proporciona que es muy interesante cuando se empieza a programar con Weka. Otra buena fuente es el libro escrito por Ian H. Witten y Eibe Frank (2005) titulado “Data Mining: Practical machine learning tools and techniques”, 2nd Edition, Morgan Kaufmann, San Francisco, 2005. Por último, otro consejo es el de partir de algoritmo ya implementado en Weka para aprender el uso que hace de los diferentes objetos y métodos y modificarlo según nuestros fines.

2. Preprocesamiento de datos

Como habéis estudiado en las clases teóricas, esta fase dentro del proceso KDD incluye las tareas de integración de datos, limpieza, reducción de la dimensionalidad, transformación de los datos, recuperación/imputación de valores perdidos, posible construcción de atributos, etc.

Por ello, al tratarse esta de la primera fase del proceso de descubrimiento de conocimiento, será la primera tarea que realicéis. Para ello se os proporciona una base de datos con un total de 5000 instancias/casos/ejemplos/filas y con 15000 características/variables/atributos/columnas

(los 14740 primeros son numéricos y el resto nominales). La variable 15001 sólo obtiene los valores -1 (casos negativos) y 1 (casos positivos).

2.1. Presentación del problema

En clase de Minería de Datos se os darán a conocer las diferentes fases del proceso de KDD y, para cada fase, aprenderéis una amplia gama de algoritmos de preprocesamiento (limpieza, discretización, reemplazo, balanceado,...) . En la comunidad científica se publican constantemente nuevas técnicas o mejoras de las ya existentes, y a menudo su rendimiento se evalúa sobre bases de datos *escogidas* o *convenientes*. Por eso, anualmente se organizan competiciones internacionales donde se entrega una base de datos proveniente de algún caso real, y el objetivo de los participantes es aplicar de forma conveniente técnicas de KDD para obtener el mejor resultado posible.

Una de las competiciones anuales más famosas es la **KDD-Cup**. La KDD-Cup 2009

<http://www.kddcup-orange.com/>

presentó como reto una base de datos de alta dimensionalidad y alto número de instancias proveniente de la compañía de telefonía Orange, donde cada registro representa a un cliente con 15000 atributos. El objetivo era intentar predecir si el cliente:

1. Se cambiará de compañía.
2. Tiende a comprar nuevos productos ofertados por la compañía.
3. Tiende a comprar productos o servicios complementarios.

En estas prácticas nos centraremos sólo en el primer objetivo; es decir, la variable clase indicará si el cliente acabó dejando Orange o no. Las prácticas también se enfocarán como una competición, de forma que los 3 estudiantes que obtengan los mejores resultados obtendrán nota extra, a establecer por los profesores de la asignatura.

Todos comenzáis con los mismos conocimientos y herramientas. Ahora todo depende de vuestro esfuerzo y lo buen estrategias que seáis!!

2.2. Trabajo a desarrollar

Esta práctica puede dividirse en fases: **1) Conversión de la base de datos al formato necesario** (En este caso, a .arff).

La base de datos que se os proporcionará será en formato CSV (*DB.csv*). Puesto que tiene 15000 atributos y 5000 instancias, no es posible cargarla en Weka en un ordenador de especificaciones técnicas normales. Por ello, primero tendréis que pensar un modo de reducirla de forma que pueda utilizarse en weka (por ejemplo, crear un script que elimine o mezcle instancias cuya variable clase sea un caso negativo, eliminar atributos nominales con un sólo estadom,...). Una vez reducida, usando Weka fácilmente la podréis convertir a .arff (abrir el csv y guardar como arff). Una vez realizado esto, guardadla con el nombre **DB.arff**. Esta será vuestra base de datos original, sobre la que aplicaréis los preprocesamientos como se explica en el paso 2.

Cuidado al obtener el .arff, pues si entre los atributos nominales alguno sólo toma valores formados por cifras y ninguna letra, pensará que es numéricos. Deberéis arreglar esto oportunamente, sobretodo en la variable clase, la cual tiene valores -1 y 1 por lo que también sufre de este problema.

2) Preprocesamiento de DB.arff

Puesto que el objetivo de la práctica es aplicar las técnicas de preprocesamiento de los datos estudiadas en clase, en concreto es obligatorio que realicéis:

- Discretización de los atributos numéricos.
- Selección de variables

Sólo para esos dos preprocesamientos Weka ofrece una gran cantidad de métodos y depende de vosotros cuántas pruebas hacer y cómo combinarlos. Además, si queréis obtener mejores resultados, es muy aconsejable

- Reemplazo de valores perdidos.
- Sobremuestrear o eliminar instancias (oversample & undersample).
- Construcción de atributos.

Puesto que tenemos una variable clase, este conjunto de datos está claramente destinado a una tarea de clasificación, por ello las técnicas que emplearéis han de ser en todo caso **supervisadas**, es decir la discretización será supervisada y la selección de variables se hará teniendo en cuenta para cada instancia cuál es el valor de clase.

Para no tener que aplicar un método de preprocesamiento cada vez que se quiere concatenar con otro, es conveniente que guardéis varias versiones de vuestro DB.arff para cada filtro o conjunto de filtros. Por ejemplo, si a DB.arff le aplicáis discretización supervisada y selección de 1000 atributos por InfoGain, en la ventana *Preprocess* podéis usar el botón *Save* para guardar la base de datos preprocesada con otro nombre, por ejemplo, DB-Disc-1000IG.arff.

Cuidado: Tened en cuenta que la pestaña *Select Attributes* sólo sirve para VER la selección, pero esta no se aplica al .arff. Para ello, debéis realizar la selección usando el filtro *SelectAttributes* en la ventana de preprocesamiento.

A la hora de seleccionar los atributos hay dos elecciones que realizar:

- **Attribute evaluator.** Lo que indicaremos aquí es la medida a usar durante la selección. Podemos optar por medidas tipo *filter* (InfoGain, Relief, ChiSquared, ...) o por una selección tipo *wrapper* (usando ID3 o NaïveBayes como evaluador). Usa la opción *More* para tener más información sobre los distintos métodos.
- **Search Method.** Indicará el tipo de búsqueda a realizar: ordenar (ranker), forward, random, genetic,

Ten en cuenta que no todo método de búsqueda combina con cualquier medida. Además, debes considerar la complejidad del proceso.

También podéis programar vuestro propio script con una técnica de selección de variables diseñada por vosotros, si razonáis debidamente la motivación para ello.

3) Evaluación Teneis que realizar una comparativa (tiempo+acierto) para las mejores opciones elegidas. Para poder comprobar la calidad del preproceso realizado, tendremos que poder evaluarlo de alguna manera. Por ejemplo si construir un clasificador Naive Bayes a partir de todos los atributos con una tasa de acierto menor que si empleamos únicamente 20 atributos seleccionados, significará por un lado que esta reducción de la dimensionalidad ha mejorado la calidad de los datos a ser tratados con Minería de Datos. Pero además, simplificaremos el proceso de aprendizaje, puesto que se ha aprendido un clasificador con 20 (+ la clase) atributos en lugar de 500!!

Para poder hacer estas comparativas, se os pide que construyais clasificadores sencillos. Se os pide que en este caso trabajéis con las siguientes variantes:

- Naive Bayes
- ID3
- C4.5 (J48 en Weka)

Para esta tarea de validación del preprocesamiento, tendréis que ir a la pestaña Classify. Como ya se ha indicado se usarán ID3, C4.5 (J48) y Naïve Bayes (opciones por defecto). En la ventana de test-options se usará *Cross Validation* o validación cruzada (5-CV). Esto lo que hace es partir la base de datos en 5 regiones y repetir el proceso 5 veces, usando en la iteración i -ésima la región i para test y el resto para data. El resultado aparece promediado.

Los resultados de la evaluación muestran varias métricas (Accuracy, recall, precision y AUC). En este caso la métrica a mejorar es AUC del estado de la clase que se refiere a los casos positivos (el estado cuyo valor es 1), es decir, cuando la etiqueta indica que el cliente de telefonía termina dejando la compañía.

4) Ganadores de la competición Cuando falte 1 semana para la presentación de la memoria, es os dará otra base de datos test.CSV. con el mismo formato que la primera que se os dió al iniciar las prácticas. Deberéis entonces:

1. Seleccionar el .arff preprocesado que mejor resultado os han dado al evaluar con el 5CV. Llamadlo train.arff
2. Reducir test.CSV de igual forma que hicisteis con la base de datos original, y crear así *test.arff*. Entonces, aplicadle la misma cadena de preprocesamientos que a train.arff.
3. Ahora que el train y test tienen el mismo formato, entrenad con el train y testear con el test usando el clasificador deseado.
4. Los resultados de los ganadores serán comprobados por los profesores de prácticas siguiendo la misma cadena de preprocesamiento para comprobar que no se han falsificado los resultados o la base de datos.

3. ¿Qué hay que hacer?

1. Aplicando los algoritmos de Weka o algún script propio, preprocesar los datos de forma que obtengáis la base de datos preprocesada (deberéis aplicar distintas técnicas y variantes, probando hasta que encontréis una o varias razonablemente buenas).
2. Es necesario realizar una comparativa donde tendréis que hacer **énfasis en el preprocesamiento**. Para ello, debéis entregar una tabla para cada clasificador distinto estudiando los preprocesamientos obtenidos y analizando su calidad. Así, las entradas en cada tabla corresponderán a preprocesamientos distintos (encadenamiento de discretizaciones diferentes con selecciones de variables diferentes). Esto os servirá para analizar si los preprocesamientos son universales o si existe alguna diferencia en los resultados si estos son empleados por clasificadores diferentes.
3. Teniendo en cuenta los resultados, intenta identificar cuáles son las variables más significativas para la tarea de clasificación.

4. ¿Qué hay que entregar?

Antes de la fecha tope, tendréis que enviar a través de moodle en la tarea correspondiente un fichero comprimido que contenga por un lado train y test (en la carpeta /preprocess). Las bases de datos deberán ser las usadas en la práctica y sus nombres deben ser significativos comenzando por train o test (o estar detallados en un fichero leeme.txt). Por otro lado una memoria explicativa del trabajo realizado (carpeta /memoria). Si además habéis programado vuestro propio código, tendréis que entregar los fuentes en java en la carpeta /codigo y los scripts en la carpeta /ejecutables.

El contenido de la memoria en lo relativo a esta práctica podría responder al siguiente índice:

1. Descripción del preproceso llevado a cabo con los datos.
2. Descripción y estudio del proceso realizado.
3. Tablas con comparativas
4. Proceso de identificación de las variables más relevantes

5. Evaluación

La evaluación de esta práctica se hará sobre la memoria de realización que cada grupo de los formados en clase deberá presentar. El contenido de dicho documento será la descripción de los pasos seguidos (sirviendo como guía orientativa lo que se os dice en el apartado anterior), las decisiones que se han tomado en cada caso razonando el por qué e interpretación de los resultados cuando proceda.

La nota de esta práctica supone **un tercio de la nota final** de prácticas, sin tener en cuenta las posibles puntuaciones extra que pueden proponerse en prácticas futuras. La valoración estará basada en la claridad de exposición y organización de la memoria, la amplitud de problemas presentes en la base de datos tratados y la solución por la que se opta.