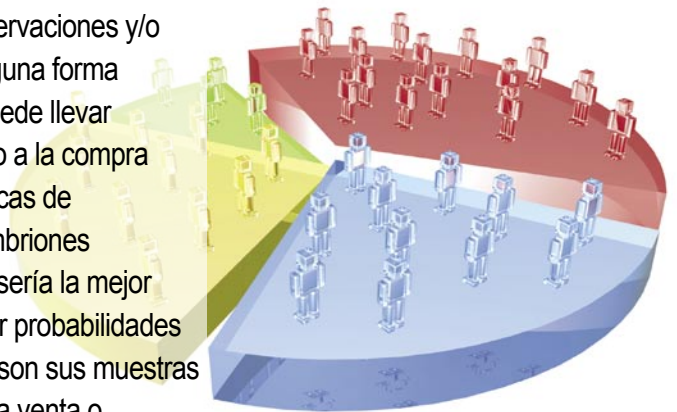




Minería de Datos

Erika Vilches González e Iván A. Escobar Broitman

En la actualidad, muchas de las decisiones importantes que se toman alrededor del mundo se basan en observaciones y/o eventos que han sido previamente registrados de alguna forma en una base o modelo de datos. Esta información puede llevar a analistas de mercado a tomar decisiones en cuanto a la compra o venta de acciones, a médicos que trabajan en clínicas de fertilización a decidir entre diferentes muestras de embriones cuyas sesenta propiedades pueden determinar cual sería la mejor opción para transferirlos al útero de una mujer y tener probabilidades exitosas de embarazo y a granjeros a decidir cuales son sus muestras de ganado más útiles a preservar antes de la próxima venta o trueque.



linux@software.com.pl

En la gran mayoría de las situaciones, los datos que se llegan a almacenar pueden contener demasiadas propiedades o atributos que causan que la información sea complicada de visualizar a primera instancia. En otros casos estas bases de datos pueden llegar a almacenar miles o millones de instancias de datos, las cuales pueden llegar a variar después de cientos o miles de muestras.

Muchos de estos datos pueden llegar a pasar desapercibidos por el ser humano y a estar siempre presentes en las más difíciles y críticas situaciones. ¿Cómo reconocer información que a plena vista no se puede distinguir? ¿Qué técnicas podemos utilizar para tratar de distinguir esta información de nuestras bases de datos?

Minería de datos

La minería de datos es la extracción de información implícita, desconocida o previamente ignorada, que puede ser potencialmente útil, de un conjunto de datos. Se puede considerar a la minería de datos como una colección de diferentes técnicas que sirven para inducir el cono-

cimiento e información de una manera estructurada de un gran conjunto de datos. A la minería de datos se le conoce en inglés como Data Mining y también se le relaciona con el descubrimiento del conocimiento en bases de datos conocido como Knowledge Data Discovery (KDD).

La minería de datos tiene una incidencia en diferentes disciplinas como la estadística, la inteligencia artificial, los aprendizajes de máquina, el reconocimiento de patrones, etc. Ésta se basa en diferentes tipos de técnicas como redes neuronales artificiales, árboles de decisión, algoritmos genéticos, el método del vecino más cercano y las reglas de inducción, entre otras.

Algunos autores definen a la minería de datos como el proceso de extraer información válida, novedosa, comprensible y potencialmente útil de un conjunto de datos. Si bien es importante descubrir esta información de un conjunto de datos, también es muy importante tener claro cuál es el significado de lo que estamos buscando para poder interpretar de una manera más eficiente la información. No sólo es importante encontrar nueva información en nuestras bases de datos, sino que tam-



bién nos es de mucha utilidad comprender los nuevos avances y resultados.

Para ello debemos tomar las siguientes consideraciones. Cuando hablamos de información, estamos hablando de diferentes niveles de datos, los cuales se muestran a continuación:

- Datos: nivel básico, contenido bruto.
- Información: manipulación de variables.
- Conocimiento: atribución de causas.
- Sabiduría: saber entender el conocimiento.

Una vez que tenemos claro el nivel de nuestros datos debemos buscar la validez de los mismos. Al momento de hablar de validez en un conjunto de datos, debemos considerar el nivel de certidumbre de la información. Tenemos que considerar la forma en la cual está representada la información y buscar la formalidad de la misma. Mientras la información que nosotros estamos analizando tenga un grado menor de formalidad, podrán haber inconsistencias tanto en los datos como las relaciones entre ellos.

Cuando tenemos bien identificados nuestros datos y su respectiva validez, tenemos que considerar que tan novedosa

o interesante puede ser la información que los mismos datos arrojan. Es muy importante cuando hablamos de minería de datos, que se considere que la información que va a presentar el resultado de dicha minería tenga un grado de novedad. Lo que normalmente se busca en estos casos es encontrar información que previamente era desconocida. Para ello debemos tomar en cuenta las verdades universales que están definidas por la sociedad y la materia en la cual se está aplicando minería de datos, y agregar los conceptos que el ser humano puede evaluar. Con ello podemos dictaminar el grado de innovación en la información extraída.

Finalmente lo más importante que debemos tomar en cuenta cuando aplicamos minería de datos a un problema, es ver que nuestros resultados sean comprensibles y útiles. La información obtenida a partir de nuestra base de datos y mediante el uso de las diferentes técnicas que conforman la minería de datos, debe ser legible al usuario. El usuario debe poder interpretar dicha información y encontrarle algún tipo de utilidad. Si la información aporta algo, que previamente no se tomaba a consideración, se puede decir que la minería de datos, fue exitosa. Si esta nueva información no aporta nada, se podrá considerar que ya se conocía todo lo posible de un conjunto de datos.

Para concluir, vale la pena mencionar que la minería de datos provee una gran utilidad a cualquier persona que tenga como fuente un conjunto de datos bien estructurado, organizado y que esté almacenado en una base de datos. Mientras mayor número de información se tenga para trabajar, mejores resultados proveerá la minería de datos.

Aplicaciones

La minería de datos es utilizada actualmente para deducir y encontrar perfiles del comportamiento de clientes, proveedores o ambientes de acuerdo a los parámetros emitidos en los modelos matemáticos que se extraen en el análisis, hecho previamente a la implementación de esta tecnología.

Un ejemplo clásico de la aplicación de minería de datos que podemos citar, es el de la detección de hábitos de compra en supermercados. Al aplicar las técnicas de minería de datos, un supermercado fue capaz de observar que había un gran número de clientes que compraban pañales y cerveza al mismo tiempo todos los días viernes. Al

estudiar este hecho, se concluyó que había una gran cantidad de padres jóvenes cuyas expectativas para el fin de semana eran quedarse en casa cuidando de sus hijos viendo la televisión y consumiendo cerveza. Debido a este estudio el supermercado pudo incrementar las ventas de cervezas aún más, al colocarlas cerca de los pañales provocando ventas compulsivas.

Un ejemplo más claro de la aplicación de minería de datos es la detección de patrones de fuga, como pueden ser en la banca y la industria de las telecomunicaciones, entre otras. La minería de datos les ayuda a estas empresas a determinar qué clientes son más factibles a darse de baja estudiando sus patrones de comportamiento y comparándolos con muestras de clientes, que efectivamente se dieron de baja en el pasado.

Por otro lado la minería de datos también se puede aplicar a investigaciones puramente científicas para obtener información que no se conocía previamente de un conjunto de datos. Tal es el caso de aplicar la minería de datos al reconocimiento acústico de aves. Se han hecho experimentos donde se utiliza la extracción de características del canto de las aves para aplicar algoritmos computacionales del área de inteligencia artificial, para distinguir entre diversos tipos de aves. Estos procesos son lentos y consumen una gran cantidad de recursos. Al aplicar la minería de datos, se pueden detectar los atributos de la señal acústica que sirven mejor para la clasificación, y utilizar solamente éstos para dicha tarea. De esta forma, se reduce el costo computacional y se acelera el procedimiento.

Software Disponible

Existe una variedad de paquetes de software para realizar Minería de Datos en Ubuntu/Linux. Dentro de los más representativos de fuente abierta y gratuitos, están los siguientes.

Weka

Del inglés *Waikato Environment for Knowledge Analysis*: Es una colección de algoritmos de aprendizaje por computadora o ML (del inglés *Machine Learning*) para realizar tareas de Minería de Datos. Los algoritmos pueden ser aplicados directamente a un conjunto de datos o *dataset* desde la interfaz gráfica del programa (Java Swing), mandándolos llamar desde el shell o utilizar los códigos independientes que se proporcionan mandándolos llamar desde nuestro



Figura 1. Ventana inicial de selección de Weka

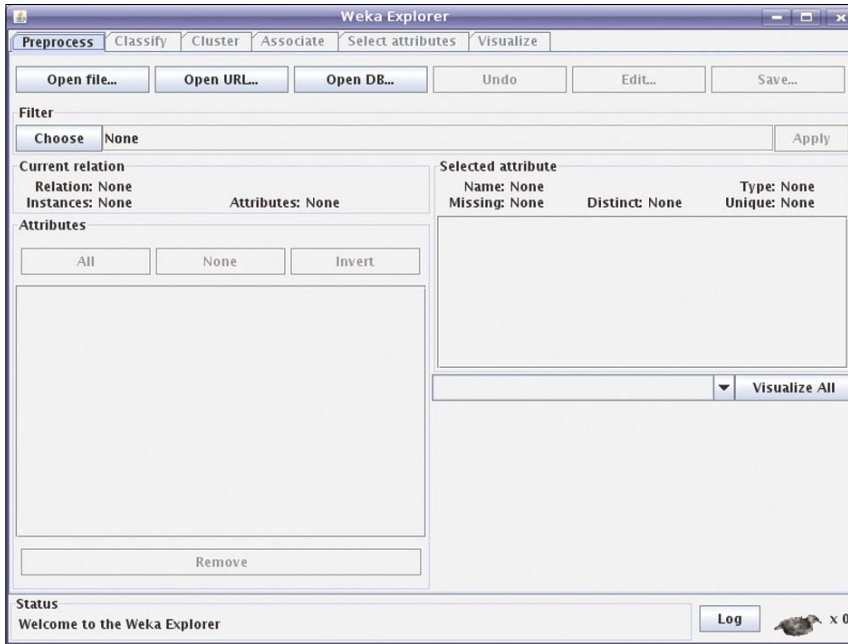


Figura 2. Interfaz gráfica de Weka

propio programa en Java, utilizándolos de la forma en que indica la documentación. Es posible escribir tus propios códigos para extender la funcionalidad del paquete ya que proporcionan el código fuente completo. Es desarrollado en la Universidad de Waikato y es un paquete de fuente abierta bajo la licencia GPL.

Los autores de este paquete, son autores también del libro "Data Mining: Practical Machine Learning Tools and Techniques", en donde además de explicarse la mayoría de

los algoritmos utilizados para realizar Minería de Datos y cómo utilizarlos para conseguir la información que se desea, se explica cómo utilizar la implementación de Weka para conseguir nuestros fines, tanto con la interfaz gráfica que proporciona el paquete, como a través de los códigos fuente.

Este paquete tiene una interfaz para la importación de datos que puede importar la información ya sea de archivos en su formato ARFF (*.arff, del inglés *Attribute Relation*

File Format), de instancias binarias serializadas, de archivos C45, de archivos separados por comas (*.csv), de archivos separados por tabulaciones utilizando el conversor que nos ofrece, de una URL o de una base de datos PostgreSQL, MySQL u Oracle mediante su JDBC. Su última versión, la 3.4.10, es del 25 de enero de 2007. Weka es el proyecto de este tipo más antiguo (se inició alrededor del año 1993) y es de los más difundidos a la fecha.

Este paquete se distribuye como un archivo comprimido, en donde se incluye la documentación, un tutorial, el log de cambios, el icono, los códigos fuente, entre otras muchas cosas. Para correr el programa se requiere tener el JRE (del inglés *Java Runtime Environment*) de Sun instalado, descomprimir el archivo (de forma que el archivo weka.jar sea accesible), y correr desde el shell `java -jar weka.jar`. (Instrucciones detalladas de instalación y uso en la sección "Manos a la obra: Minando una Base de Datos con Weka").

URL: <http://www.cs.waikato.ac.nz/ml/weka/>

YALE

Del inglés *Yet Another Learning Environment*: Es una colección de operadores de Minería de Datos (más de 400) desarrollada en Java que integra completamente los códigos de Weka y que nos permite realizar ML y Minería de Datos. Cuenta con una interfaz gráfica fácil de utilizar, pero también puede ser utilizado desde el shell o como una librería dentro de tus propios programas en Java. Para instalarlo, se requiere tener instalado con anterioridad el JRE de Sun (ver procedimiento de instalación en la sección "Manos a la obra: Minando una Base de Datos con Weka"), bajar el archivo `yale-3.4-bin.zip`, descomprimirlo y correr el archivo `lib/yale.jar` que viene dentro del paquete con el comando `java -jar lib/yale.jar` desde el shell.

YALE cuenta con un mecanismo sencillo para desarrollar extensiones y *plugins* que hace posible integrar nuevos operadores y con ello adaptar el paquete para los requerimientos personales. Existen *plugins* ya desarrollados disponibles en la página de YALE: *Clustering Plugin*, *Word Vector Tool Plugin*, *Value Series Plugin*, *Distributed Data Mining Plugin*, *Data Stream Plugin* y *Selur-tib8 Plugin* (un juego de los 80s).

Este paquete tiene dos tipos de licencia, la gratuita bajo la licencia GPL y la propietaria, para cuando se desea modificar YALE y distribuirlo sin proporcionar los códigos

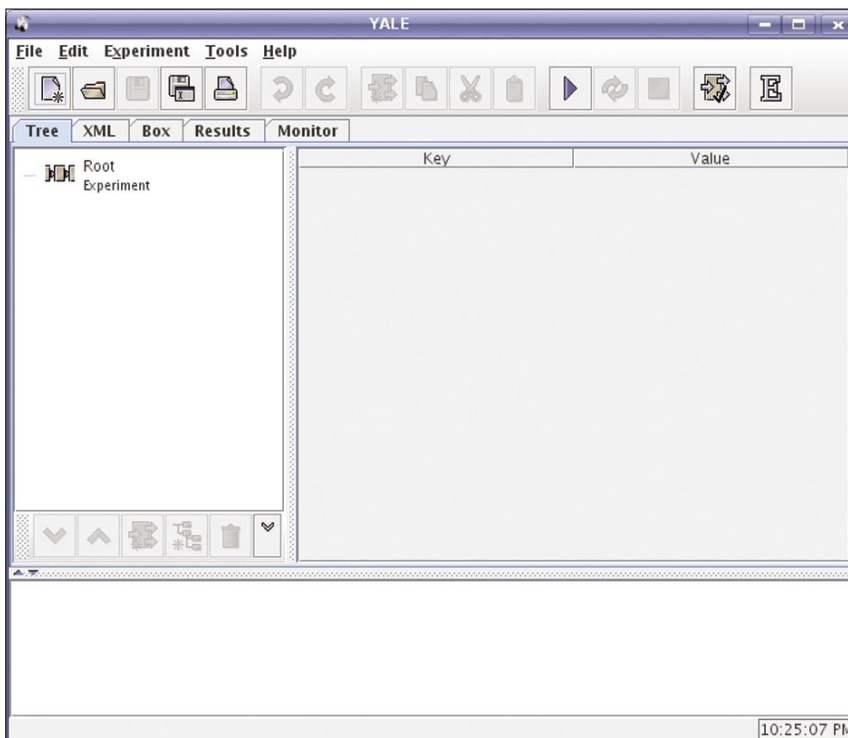


Figura 3. Interfaz gráfica de YALE



fuente, desarrollar módulos comerciales extra, proporcionar servicio al cliente, soluciones individuales o soporte profesional.

YALE puede importar información a partir de distintos formatos de archivos como ARFF, csv, C45, SVMLight, mySVM, Excel, SPSS, etc., archivos de texto (a través del Word Vector Plugin), archivos de audio (a través del Value Series Plugin) o a partir de sistemas de bases de datos como Oracle, MySQL, PostgreSQL, Microsoft SQL Server, Sybase, dBase, etc., e inclusive es capaz de trabajar directamente sobre una base de datos. Su última versión, la 3.4, es del 3 de octubre de 2006. El proyecto YALE se inició alrededor del año 2001, y comparado con su antecesor Weka, es más flexible en la realización de experimentos y cuenta con más operadores disponibles, por lo que está ganando popularidad rápidamente a pesar de los años que le lleva de ventaja Weka.

URL: <http://rapid-i.com/>

R: R

Es otra alternativa de fuente abierta con desarrollo activo y una gran comunidad de usuarios. Para R, se encuentra disponible Rattle (del inglés *R Analytical Tool to Learn Easily*), herramienta gratuita y de fuente abierta (bajo la licencia GPL) escrita en el lenguaje estadístico R utilizando la interfaz gráfica de Gnome. Soporta una colección creciente de algoritmos que pueden ser utilizados para realizar Minería de Datos y provee una interfaz intuitiva que guía al usuario a través de los pasos necesarios. La última versión de Rattle es la 2.2.56 (revisión 188), que fue liberada el día 25 de abril de 2007.

Para instalar Rattle, se requiere lo siguiente:

- Las librerías GTK+, que proveen la interfaz de usuario Gnome utilizada por Rattle. Si se utiliza Gnome, como en el caso de Ubuntu, éstas ya se encuentran instaladas, y solo es necesario asegurarse de que el paquete Glade también se encuentre instalado (`sudo apt-get install lib-glade2-0` desde el shell). Si no se utiliza Gnome, como en el caso de Kubuntu, es necesario instalar las librerías GTK+ (`sudo apt-get install libgtk2.0-0` desde el shell).
- El paquete estadístico R (`sudo apt-get install r-base` desde el shell)
- RGtk2 (`sudo apt-get install r-cran-rgtk2` desde el shell)

Listado 1. Archivo sources.list modelo que utiliza los repositorios locales de España

```
## See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
## newer versions of the distribution.
## Add comments (##) in front of any line to remove it from being checked.
## Use the following sources.list at your own risk.
## Uncomment deb-src if you wish to download the source packages
## If you have a install CD you can add it to the repository using 'apt-cdrom
add'
## which will add a line similar to the following:
#deb cdrom:[Ubuntu 7.04 _Feisty Fawn_ - Beta i386 (20070322.1)]/ feisty main
restricted
deb http://es.archive.ubuntu.com/ubuntu/ feisty main restricted
#deb-src http://es.archive.ubuntu.com/ubuntu/ feisty main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://es.archive.ubuntu.com/ubuntu/ feisty-updates main restricted
#deb-src http://es.archive.ubuntu.com/ubuntu/ feisty-updates main restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://es.archive.ubuntu.com/ubuntu/ feisty universe
#deb-src http://es.archive.ubuntu.com/ubuntu/ feisty universe
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://es.archive.ubuntu.com/ubuntu/ feisty multiverse
#deb-src http://es.archive.ubuntu.com/ubuntu/ feisty multiverse
## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://es.archive.ubuntu.com/ubuntu/ feisty-backports main restricted
universe multiverse
#deb-src http://es.archive.ubuntu.com/ubuntu/ feisty-backports main
restricted universe multiverse
deb http://security.ubuntu.com/ubuntu feisty-security main restricted
#deb-src http://security.ubuntu.com/ubuntu feisty-security main restricted
deb http://security.ubuntu.com/ubuntu feisty-security universe
#deb-src http://security.ubuntu.com/ubuntu feisty-security universe
deb http://security.ubuntu.com/ubuntu feisty-security multiverse
#deb-src http://security.ubuntu.com/ubuntu feisty-security multiverse
## PLF REPOSITORY (Unsupported. May contain illegal packages. Use at own
risk.)
## Medibuntu - Ubuntu 7.04 "feisty fawn"
## Please report any bug on https://launchpad.net/products/medibuntu/+bugs
deb http://medibuntu.sos-sts.com/repo/ feisty free non-free
#deb-src http://medibuntu.sos-sts.com/repo/ feisty free non-free
## CANONICAL COMMERCIAL REPOSITORY (Hosted on Canonical servers, not Ubuntu
## servers. RealPlayer10, Opera, DesktopSecure and more to come.)
deb http://archive.canonical.com/ubuntu feisty-commercial main
```

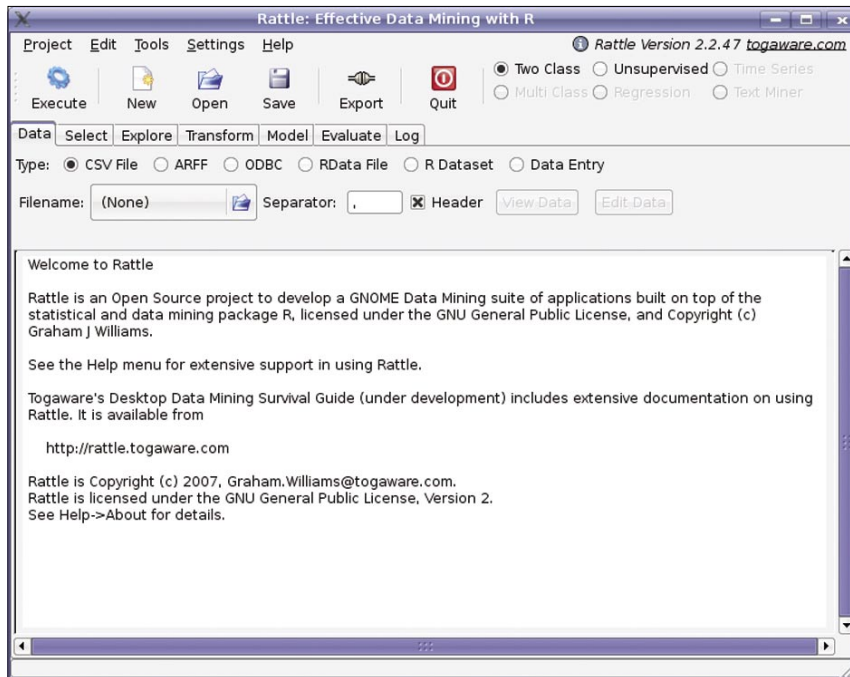


Figura 4. Rattle, Minería de Datos con R

- Los paquetes adicionales de R (iniciar R escribiendo "sudo R" en el shell y desde dentro de R "install.packages(rattle, dependencies=TRUE)")
- Rattle ("install.packages(rattle)" desde dentro de R)

Para utilizar Rattle es necesario iniciar R ("R" desde el shell), cargar el paquete rattle ("library(rattle)" desde R) e iniciar la interfaz gráfica de Rattle (opcional, "rattle()" desde R).

Como parte de este proyecto, se encuentra disponible gratuitamente el libro electrónico "Data Mining Desktop Survival Guide" <http://datamining.togaware.com/survivor/index.html>, que explica los conceptos y algoritmos de la Minería de Datos ilustrándolos con ejemplos escritos en R. Este libro se actualiza constantemente y su última versión es del 24 de abril de 2007.

URL: <http://rattle.togaware.com>

Manos a la obra: Minando una Base de Datos con Weka

Para hacer una pequeña demostración de lo que la Minería de Datos es capaz de hacer, seleccionamos a Weka como nuestro paquete de elección por ser el más ampliamente difundido. Como habíamos comentado anteriormente, para utilizar Weka es necesario tener primero instalado el JRE de Sun, sin embargo para instalarlo es necesario poder instalar software disponible en el repositorio multiverse. A conti-

nuación describiremos los pasos necesarios para tener todo funcionando como se requiere en Kubuntu Feisty Fawn, asumiendo que nuestra instalación es nueva y que no se le ha hecho ninguna modificación:

Añadir Repositorios Adicionales

- Crear un respaldo del archivo que contiene nuestra lista de fuentes: `sudo cp -p /etc/apt/sources.list /etc/apt/sources.list_backup` desde el shell,

- Abrir la lista de fuentes en un editor de texto: `sudo kate /etc/apt/sources.list` desde el shell,
- Utilizar como modelo el contenido del ejemplo siguiente, que utiliza los repositorios locales de España (en caso de que se desee utilizar los de otro país cambiar "es" por el código del país en las ocurrencias de `es.archive.ubuntu.com`), Listado 1,
- Salvar el archivo,
- Bajar las llaves GPG necesarias. Ejemplo: para las llaves GPG del repositorio PLF utilizar `"wget -q http://packages.medibuntu.org/medibuntu-key.gpg -O- | sudo apt-key add -"` desde el shell,
- Refrescar la lista de paquetes: `sudo aptitude update` desde el shell.

Instalar el JRE de Sun

La versión de Java que viene instalada por defecto con Kubuntu Feisty Fawn es la `gij 4.1.2 (GNU libgcj)`, lo que se puede confirmar con el comando `"java -version"` desde el shell. Esta versión no nos sirve para correr Weka, ya que nos marca el siguiente error indicando que no puede cargar el AWT toolkit: "Exception in thread "main" java.awt.AWTError: Cannot load AWT toolkit: gnu.java.awt.peer.gtk.GtkToolkit". Es por esta razón que requerimos instalar JRE de Sun, lo que se hace de la siguiente forma:

1. `sudo aptitude install sun-java6-jre sun-java6-fonts` desde el shell,

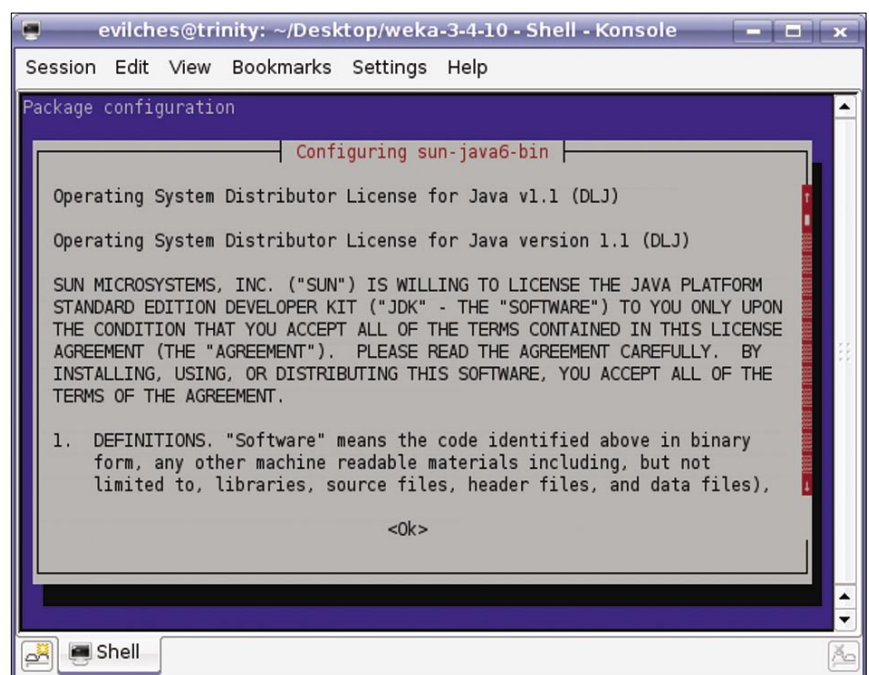


Figura 5. DLJ en la instalación del JRE de Sun



- Indicar al instalador que deseamos continuar con la instalación,
- Leer y en su caso aceptar la licencia DLJ (del inglés *Distributor Licence for Java*).

Obtener y Correr Weka (interfaz gráfica)

- En la página <http://www.cs.waikato.ac.nz/ml/weka/> ir a la sección de "Download", buscar "Other platforms (Linux, etc.)" y hacer clic en donde dice *Click here to download a zip archive containing Weka*,
- Descomprimir el archivo *weka-3-4-10.zip*.
- Desde el shell, correr el archivo *weka.jar* de la siguiente forma: `java -Xmx1024M -jar weka.jar`, lo que abre la ventana *Weka GUI Chooser* e inicia Weka con 1024MB disponibles para su uso (la base de datos que utilizaremos es demasiado grande para cargarla con la memoria que se le asigna de modo predeterminado a Weka). En esta ventana, presionar el botón *Explorer*, esto abrirá una nueva ventana llamada *Weka Explorer*.

Obtener y Cargar la Base de Datos a Minar

- Vamos a utilizar para este ejemplo la base de datos pública de evolución del cáncer de mama de *Kent Ridge Biomedical Data Set Repository* disponible en la página <http://research.i2r.a-star.edu.sg/rp/>. Esta es una base de datos de la evolución de pacientes con cáncer de mama a 5 años. El conjunto de entrenamiento (utilizado para entrenar al algoritmo de clasificación) consta de 78 muestras (pacientes), 34 de las cuales son de pacientes que desarrollaron metástasis distantes dentro de los 5 años subsiguientes al diagnóstico inicial y pertenecen a la clase "recaída" o *relapse*. Las 44 muestras restantes son de pacientes que permanecieron sin cáncer después del diagnóstico inicial por un período de al menos 5 años y pertenecen a la clase "no recaída" o *non-relapse*. El conjunto de prueba (utilizado para verificar la eficiencia del algoritmo de clasificación), consta de 12 muestras de la clase recaída y 7 de la clase no recaída. El número de genes utilizado es de 24.481 (atributos que analizará el algoritmo para determinar la manera en la que influyen a la clasificación),

Descomprimir el archivo *BreastCancer.zip*. Dentro del archivo, podemos ver los archivos *breastCancer-test.arff*, *breastCancer-train.arff* y la carpeta *BreastCancer* que contiene los archivos *breastCancer_test.data*, *breastCancer_train.data* y *breastCancer.names*.

- Desde el *Weka GUI Chooser* (la ventana principal de Weka), oprimir el botón *Open file...* y seleccionar el archivo *breastCancer-train.arff*. Esperar unos minutos a que cargue. Una vez que ha terminado de cargar, nos muestra que tenemos una base de 78 instancias (muestras) y 24.482 atributos (24.481 genes y la clase a la que pertenecen), de los que nos indica el nombre en la sección "Attributes".

Aplicar Minería de Datos a la Base de Datos

- Hacer clic en la pestaña "Classify", y posteriormente en el botón "Choose", donde escogeremos el algoritmo *Trees* -> *J48*. *J48* es la implementación de *C4.5* de Weka, pero no del algoritmo *C4.5* original, sino de una versión mejorada del mismo llamada *C4.5* revisión 8, que a su vez es una evolución del algoritmo *ID3*. Todos estos algoritmos nos permiten formar árboles de decisión para clasificar instancias, sin embargo en el caso de *C4.5* y *J48*, entre otras diferencias, se pueden utilizar instancias numéricas y no sólo nominales como en el caso de su antecesor *ID3*. Es por esta habilidad adicional que *J48* fue seleccionado para este ejemplo, debido a que nuestros atributos tienen valores numéricos,
- En la sección *Test options* seleccionar *Supplied test set* y presionar el botón *Set...*, lo que abrirá una nueva ventana donde presionaremos el botón *Open file...* y seleccionaremos el archivo *breastCancer-test.arff*. Posteriormente, cerrar la ventana *Test instances* y volver a la ventana *Weka Explorer*,
- Presionar el botón *Start* y esperar unos instantes a que corra el algoritmo, (ha terminado de correr cuando el ave de la esquina inferior derecha deja de *bailar* y permanece quieta),
- El resultado que arrojó el algoritmo lo podemos observar en la sección *Classifier output*, en donde se puede apreciar lo siguiente (Listado 2).

Listado 2. Salida del clasificador J48 en Weka

```

=== Run information ===
Scheme: weka.classifiers.trees.J48
-C 0.25 -M 2
Relation: changed by huiqing
Instances: 78
Attributes: 24482
[list of attributes omitted]
Test mode: user supplied test set:
19 instances
=== Classifier model (full training set) ===
J48 pruned tree
-----
NM_013438 <= 0.148
| AF018081 <= 0.158
| | NM_001157 <= -0.095: relapse (6.0)
| | NM_001157 > -0.095
| | | Contig40319_RC <= -0.139: relapse (4.0)
| | | Contig40319_RC > -0.139
| | | NM_012479 <= 0.16: non-relapse (43.0)
| | | NM_012479 > 0.16: relapse (4.0/1.0)
| AF018081 > 0.158: relapse (9.0)
NM_013438 > 0.148: relapse (12.0)
Number of Leaves : 6
Size of the tree : 11
Time taken to build model: 9.21 seconds
=== Evaluation on test set ===
=== Summary ===
Correctly Classified Instances 14
73.6842 %
Incorrectly Classified Instances 5
26.3158 %
Kappa statistic 0.5078
Mean absolute error 0.2632
Root mean squared error 0.513
Relative absolute error 50.9554 %
Root relative squared error 98.6602 %
Total Number of Instances 19
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall
F-Measure Class
0.583 0 1 0.583 0.737 relapse
1 0.417 0.583 1 0.737 non-relapse
=== Confusion Matrix ===
a b <-- classified as
7 5 | a = relapse
0 7 | b = non-relapse

```



Análisis de Resultados

La sección *Run information* nos indica que se aplicó el algoritmo J48 y los parámetros con los que se aplicó: sin particiones binarias, con un factor de confianza de 0.25, un número mínimo de objetos de 2, 3 dobles para la validación cruzada, con podado (recortado del árbol), etc.

La sección *Classifier model (full training set)* nos muestra primero que nada, el árbol que se construyó, el número de hojas y el tamaño del mismo (6 y 11 respectivamente, para este ejemplo) y el tiempo que se tardó en construir el modelo (9.21 segundos). El árbol resultante se interpreta de la siguiente forma: el gen NM_013438 es la raíz del árbol (es el más importante para determinar la clase a la que pertenece una muestra), si su valor es mayor que 0.148, el paciente sufrirá una recaída, si es menor o igual a ese valor, se tomará en cuenta el valor del gen AF018081: si es mayor que 0.158, el paciente sufrirá una recaída, si es menor a ese valor, se tomará en cuenta el valor del gen NM_001157: si es menor o igual a -0.095, el paciente sufrirá una recaída, mientras que si es mayor a ese valor, deberá tomarse en cuenta el valor del gen Contig40319_RC: si es menor o igual a -0.139, el paciente sufrirá una recaída, mientras que si es mayor a ese valor, deberá tomarse en cuenta el valor del gen NM_012479: si es mayor a 0.16, el paciente sufrirá una recaída, mientras que si es menor o igual a ese valor, el paciente no sufrirá de una recaída. El número entre paréntesis que se encuentra al lado de la clase,

indica el número de instancias que cayeron en esa hoja y, en su caso, está seguido de una "/" y del número de esas instancias que fueron mal clasificadas como pertenecientes a esa hoja, de modo que (4.0/1.0) se interpreta como: cayeron 4 instancias en esta hoja, de las cuales 1 está incorrectamente clasificada. La sumatoria del primer número entre paréntesis de todo el árbol nos da el total de instancias utilizadas para el entrenamiento (78 en este caso).

La sección *Summary* nos presenta el número de instancias del conjunto de prueba correctamente clasificadas y su porcentaje (14 y 73.68% respectivamente), el número de instancias del conjunto de prueba incorrectamente clasificadas y su porcentaje (5 y 26.31% respectivamente), estadísticas sobre los errores obtenidos (media del error absoluto, la raíz del error cuadrático medio, etc.) y el número total de instancias de prueba.

La sección *Detailed Accuracy By Class* nos presenta estadísticas sobre los resultados en el conjunto de prueba para cada clase: la tasa de verdaderos positivos ((verdaderos positivos / (verdaderos positivos + falsos negativos)) x 100), la tasa de falsos positivos ((falsos positivos / (falsos positivos + verdaderos negativos)) x 100), la precisión ((verdaderos positivos / (verdaderos positivos + falsos positivos)) x 100), el *recall* (es lo mismo que la tasa de verdaderos positivos) y el F-Measure ((2 x *recall* x precisión) / (*recall* + precisión)).

La sección *Confusion Matrix* nos presenta la matriz de confusión de las clases, que se interpreta de la siguiente forma: la clase a representa la recaída, mientras que la clase b representa la no recaída; la clase correcta es la fila, mientras que la clase predicha es la columna; 7 instancias de la clase a fueron correctamente

clasificadas como de la clase a, mientras que 5 instancias de la clase a fueron incorrectamente clasificadas como de la clase b; ninguna instancia de la clase b fue incorrectamente clasificada como de la clase a, mientras que 7 instancias de la clase b fueron correctamente clasificadas como de la clase b. Desde el punto de vista de la clase a, la celda (a(y),a(x)) = 7 representa a los verdaderos positivos, la celda (a(y),b(x)) = 5 representa a los falsos negativos, la celda (b(y),a(x)) = 0 representa a los falsos positivos, y la celda (b(y),b(x)) = 7 representa a los verdaderos negativos. Desde el punto de vista de la clase b, la celda (a(y),a(x)) = 7 representa a los verdaderos negativos, la celda (a(y),b(x)) = 5 representa a los falsos positivos, la celda (b(y),a(x)) = 0 representa a los falsos negativos, y la celda (b(y),b(x)) = 7 representa a los verdaderos positivos.

Lo más importante que se abstrae de estos resultados es que a los pacientes que no recayeron, se les predijo en todos los casos (100% de ellos). En el caso de los pacientes que recayeron, se les predijo a un 58% de ellos que recaerían, mientras que al 42% restante se les predijo que no recaerían pero recayeron. Este es un ejemplo del tipo de información que se puede obtener a través de la Minería de Datos, descubriendo información que se encuentra *escondida* en grandes bases de datos (que no es visible a simple vista). Los resultados pueden complementarse utilizando otros algoritmos de Minería de Datos, para enriquecer el resultado obtenido. Por ejemplo, para nuestro caso de predicción de la evolución del cáncer, podríamos utilizar el algoritmo Naïve-Bayes para conocer el porcentaje de pertenencia a cada clase de una instancia, ayudándonos a conocer qué tan certero es el diagnóstico. 📌



Sobre los autores

La Maestra Erika Vilches González estudio la carrera de Ingeniero en Sistemas Computacionales y una Maestría en Ciencias Computacionales. Actualmente, estudia el Doctorado en Ciencias Computacionales en el ITESM Campus Estado de México. Es socia fundadora de Quetzal Hosting y se encarga principalmente de la administración de servidores y el desarrollo de software a medida.

El Maestro Iván Alejandro Escobar Broitman, es profesor del departamento de Ciencias Computacionales del Tecnológico y de Estudios Superiores de Monterrey Campus Estado de México. Estudió la carrera de Ingeniería en Sistemas Electrónicos y una Maestría en Ciencias Computacionales. Actualmente estudia el Doctorado en Ciencias Computacionales y es socio fundador de Quetzal Hosting.