UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

LICENCIATURA EN INGENIERÍA EN SISTEMAS INTELIGENTES

MANUAL PARA PRÁCTICAS DEL LABORATORIO DE CÓMPUTO

ASIGNATURA:

TALLER CON WEKA

ELABORARÓN:

DRA. DORICELA GUTIÉRREZ CRUZ
M. en C. YAROSLAF AARÓN ALBARRÁN FERNÁNDEZ
DR. RICARDO RICO MOLINA

OCTUBRE 2017

MANUAL PARA PRÁCTICAS DEL LABORATORIO DE CÓMPUTO PARA LA ASIGNATURA TALLER CON WEKA

IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

Espacio académico: CENTRO UNIVERSITARIO NEZAHUALCÓYOTL								
Programa educativo: LICENCIATURA DE INGENIERÍA EN SISTEMAS INTELIGENTES					**Área de docencia:** HERRAMIENTA PARA LOS SISTEMAS INTELIGENTES			
Aprobación de los HH Consejos Académico y de Gobierno			**Fecha:** OCTUBRE 2017		**Programa elaborado por:** Doricela Gutiérrez Cruz, Yaroslaf Aarón Albarrán Fernández, Ricardo Rico Molina.			
Nombre de la unidad de aprendizaje: Taller con WEKA					**Fecha de elaboración:** AGOSTO 2017			
Clave	**Horas de Teoría**	**Horas de Práctica**	**Total de horas**	**Créditos**	**Área curricular:**	**Carácter de la unidad de aprendizaje**	**Núcleo de formación**	**Modalidad**
L40667	1.0	2.0	3.0	4.0	HERRAMIENTA PARA LOS SISTEMAS INTELIGENTES	Optativa	INTEGRAL	ESCOLARIZADA CON ADMINISTRACIÓN FLEXIBLE DE LA ENSEÑANZA
Prerrequisitos (Conocimientos previos): Minería de datos I y II			**Unidad de aprendizaje antecedente:** NINGUNA			**Unidad de aprendizaje consecuente:** NINGUNA		
Programas en los que se imparte: LICENCIATURA DE INGENIERÍA EN SISTEMAS INTELIGENTES								

EL PRESENTE MANUAL DE PRÁCTICAS HA SIDO AVALADO EN EL MES DE OCTUBRE DE 2017 POR:

Centro Universitario UAEM Nezahualcóyotl

H. CONSEJO DE GOBIERNO CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

H. CONSEJO ACADÉMICO CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

ÍNDICE

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

DIRECTORIO

Dr. en Ed. Alfredo Barrera Baca

RECTOR

M. en S. P. María Estela Delgado Maya

SECRETARIA DE DOCENCIA

Dr. en C.I.Amb. Carlos Eduardo Barrera Díaz

SECRETARIO DE INVESTIGACIÓN Y ESTUDIOS AVANZADOS

Dr. en C.S. Luis Raúl Ortiz Ramírez

SECRETARIO DE RECTORÍA

Dr. en A. José Edgar Miranda Ortiz

SECRETARIO DE DIFUSIÓN CULTURAL

M. en Com. Jannet Socorro Valero Vilchis

SECRETARIA DE EXTENSIÓN Y VINCULACIÓN

M. en E. Javier González Martínez

SECRETARIO DE ADMINISTRACIÓN

M. en E.U.R. Héctor Campos Alanís

SECRETARIO DE PLANEACIÓN Y DESARROLLO INSTITUCIONAL

M. en L.A. María del Pilar Ampudia García

SECRETARIA DE COOPERACIÓN INTERNACIONAL

Dra. en C.S. y Pol. Gabriela Fuentes Reyes

ABOGADA GENERAL

Lic. en Com. Gastón Pedraza Muñoz

DIRECTOR GENERAL DE COMUNICACIÓN UNIVERSITARIA

M. en R.I. Jorge Bernaldez García

SECRETARIO TÉCNICO DE LA RECTORÍA

M. en A.P. Guadalupe Santamaría González

DIRECTORA GENERAL DE CENTROS UNIVERSITARIOS UAEM Y UAP

M. en A. Ignacio Gutiérrez Padilla

CONTRALOR UNIVERSITARIO

CENTRO UNIVERSITARIO NEZAHUALCÓYOTL

DIRECTORIO

M. en I. S. C. Cuauhtémoc Hidalgo Cortés
Encargado del despacho de C.U. Nezahualcóyotl

Dr. en E. Darío Guadalupe Ibarra Zavala
Subdirector Académico

Lic. en E. Ramón Vital Hernández
Subdirector Administrativo

Dra. en C. S. María Luisa Quintero Soto

Coordinadora de Investigación y Estudios Avanzados

Lic. en A. E. Víctor Manuel Durán López
Coordinador de Planeación y Desarrollo Institucional

Dr. en R.I. Rafael Alberto Duran Gómez
Coordinador de la Licenciatura en Comercio Internacional

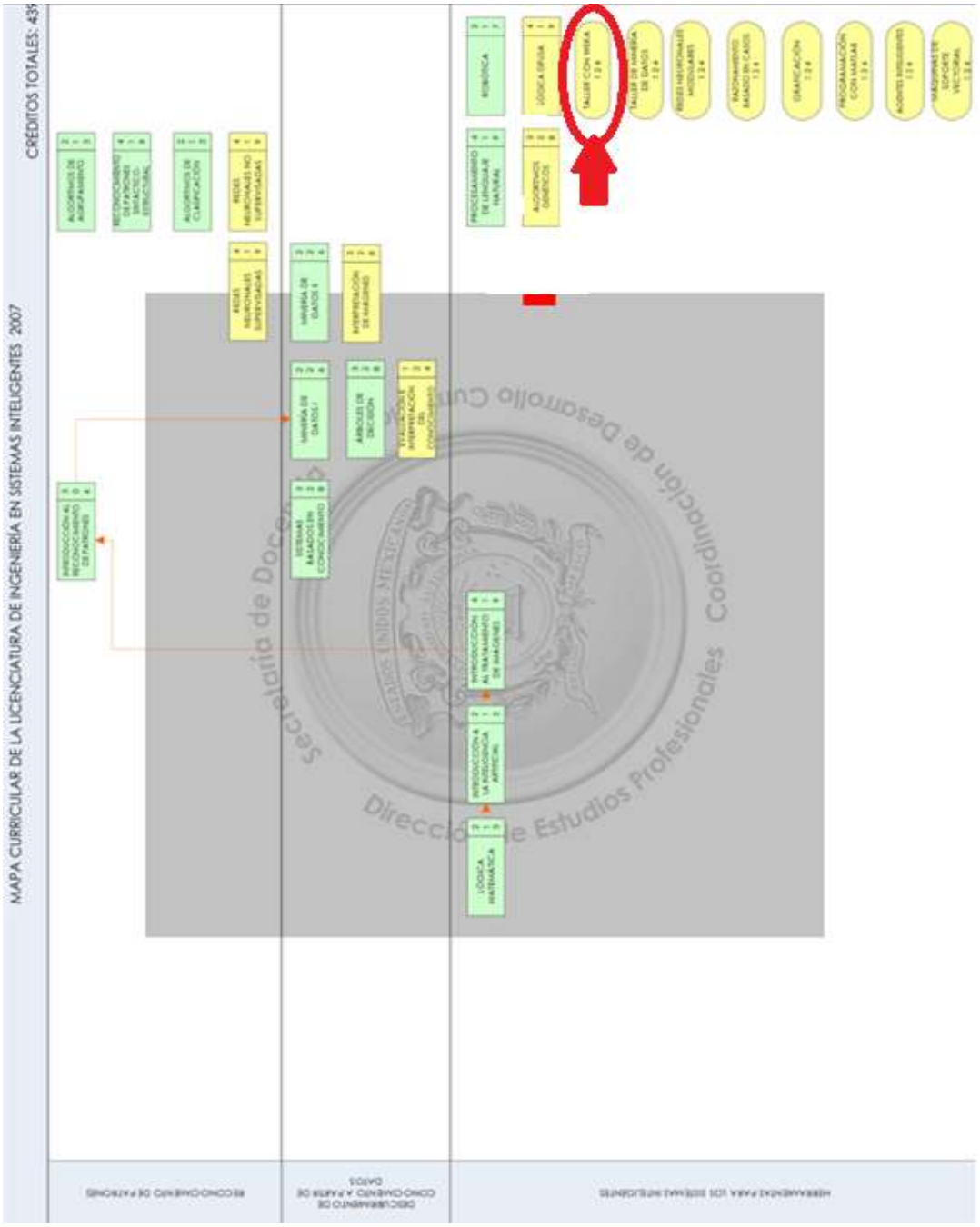
Dra. Silvia Padilla Loredo
Coordinadora de la Licenciatura en Educación para la Salud

Dra. Ricardo Rico Molina
Coordinador de la licenciatura en Ingeniería en Sistemas Inteligentes

D. En U. Noé Gaspar Sánchez
Coordinador de Ingeniería en Transporte

M. En CC Erick Nicolás Cabrera Álvarez
Coordinador de Licenciatura en Seguridad Ciudadana

Ubicación de la asignatura de Taller con Weka, dentro del programa de la Lic. en Ing. en Sistemas Inteligentes.



SECUENCIA DIDÁCTICA

PRÁCTICA 1

INSTALACIÓN DEL SOFTWARE.

PRÁCTICA 2

ENTORNO DE TRABAJO DEL SOFTWARE WEKA.

PRÁCTICA 3

PREPARACIÓN DE DATOS.

PRÁCTICA 4

TRABAJO CON FILTROS.PREPARACIÓN DE ARCHIVOS DE MUESTRA.

PRÁCTICA 5

PROCESAMIENTO DE DATOS.

PRÁCTICA 6

AGRUPAMIENTO NUMÉRICO

PRÁCTICA 1
INSTALACIÓN DEL SOFTWARE WEKA

Objetivo

El alumno aprenderá a instalar el software de WEKA que le servirá para familiarizarse con las herramientas básicas que este proporciona.

Introducción

Hoy en día numerosos sistemas de información emplean técnicas de Data Mining y Machine Learning para obtener información y conocimiento de grandes volúmenes de datos tanto internos como externos a la entidad que posee dicho sistema de información. Como es el caso de la herramienta Weka que permite realizar diferentes análisis de datos almacenados en bases de datos relacionales (JDBC), archivos CSV y archivos ARFF.

Weka (*Waikato Environment for Knowledge Analysis*, en español «entorno para análisis del conocimiento de la Universidad de Waikato») es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka es software libre distribuido bajo la licencia GNU-GPL.

Contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original de Weka fue un *front-end* en TCL/TK para modelar algoritmos implementados en otros lenguajes de programación, más unas utilidades para pre-procesamiento de datos desarrolladas en C para hacer experimentos de aprendizaje automático. Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (WEKA 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación.

Dentro de los principales aspectos relevantes de Weka, destacan: licencia pública general de GNU, portabilidad; está completamente implementado en Java y puede correr en casi

cualquier plataforma, contiene una extensa colección de técnicas para pre-procesamiento de datos y modelado, es fácil de utilizar gracias a su interfaz gráfica de usuario.
Weka soporta varias tareas estándar de minería de datos, especialmente, pre-procesamiento de datos, clustering, clasificación, regresión, visualización, y selección. Todas las técnicas de Weka se fundamentan en la asunción de que los datos están disponibles en un archivo plano (*flat file*) o una relación, en la que cada registro de datos está descrito por un número fijo de atributos (normalmente numéricos o nominales, aunque también se soportan otros tipos), también proporciona acceso a bases de datos vía SQL gracias a la conexión JDBC (*Java Database Connectivity*) y puede procesar el resultado devuelto por una consulta hecha a la base de datos.

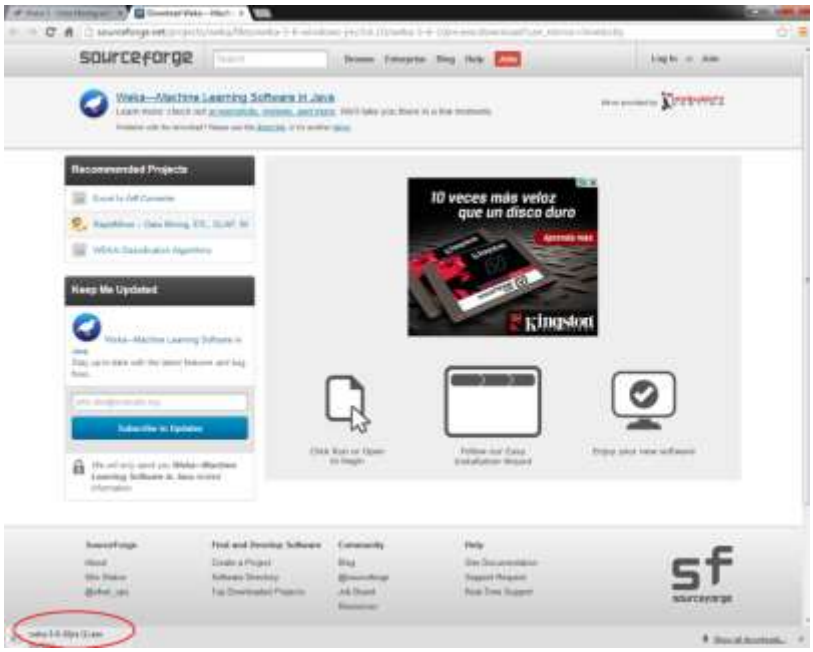
Desarrollo

A continuación se describe paso a paso del proceso para realizar la instalación de la herramienta Weka, desde la descarga del software hasta la configuración del mismo.

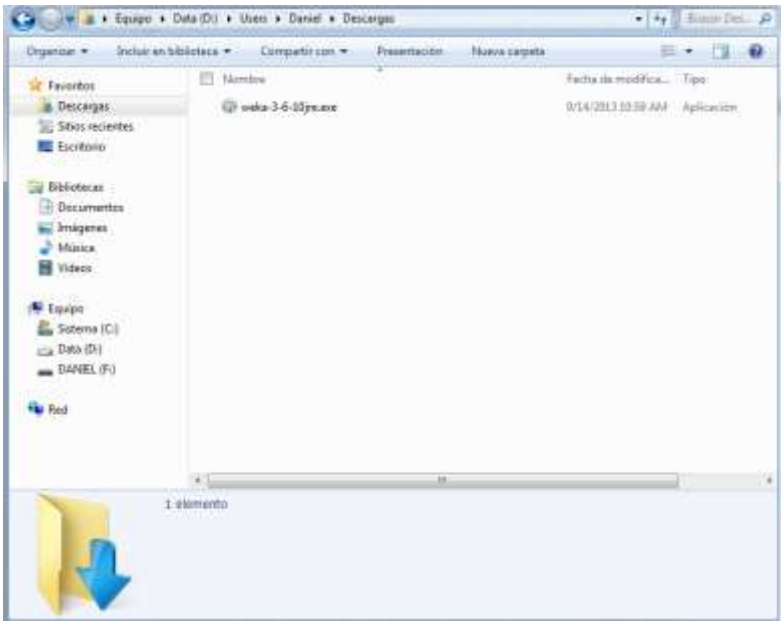
Descargar instalador Weka

Para realizar la descarga de la herramienta Weka ingrese al siguiente link: http://www.cs.waikato.ac.nz/ml/weka/downloading.html , en este podrá encontrar el software para diferentes sistemas operativos como Windows, Mac OSX y Linux.

Después de seleccionar el sistema operativo y la arquitectura (En caso de Windows x86 y x64) encontrar la siguiente página. En la parte inferior izquierda podrá observar el porcentaje de descarga de la herramienta.

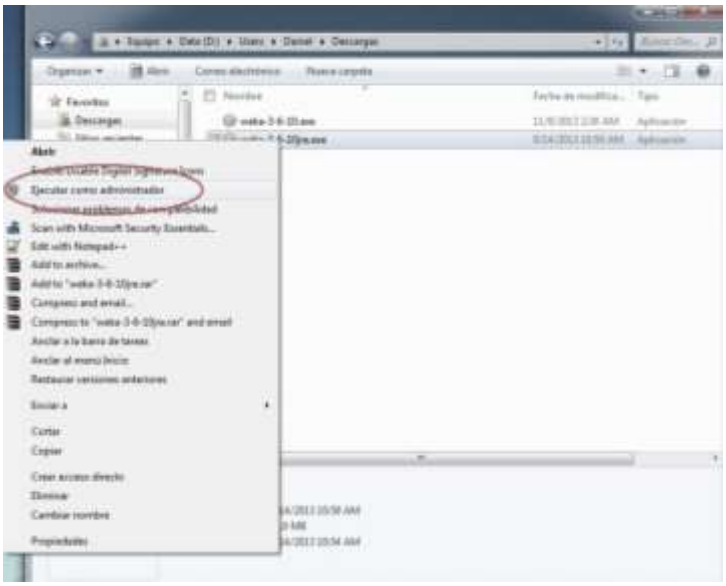


En la siguiente ventana podrá visualizar el ejecutable de la herramienta descargado.



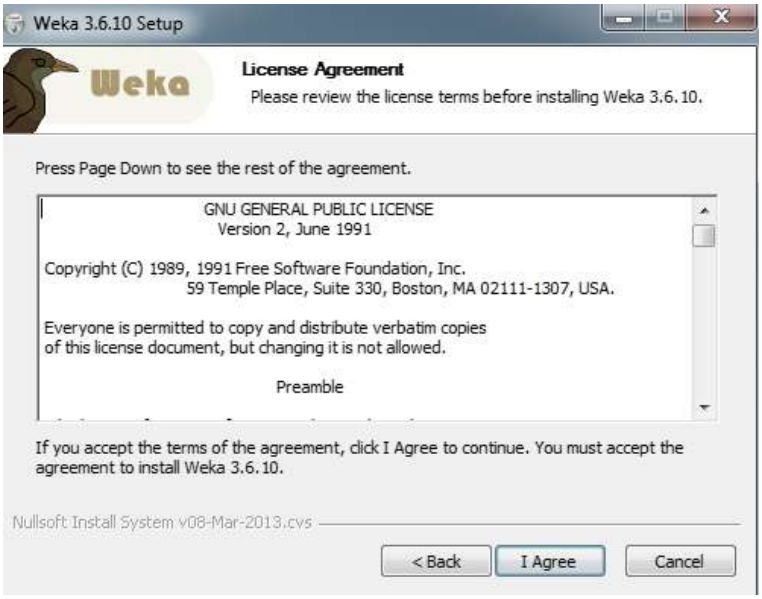
Instalación Weka

1. Hacer clic derecho sobre el instalador, seguido de un doble clic en la opción *Ejecutar como administrador*.



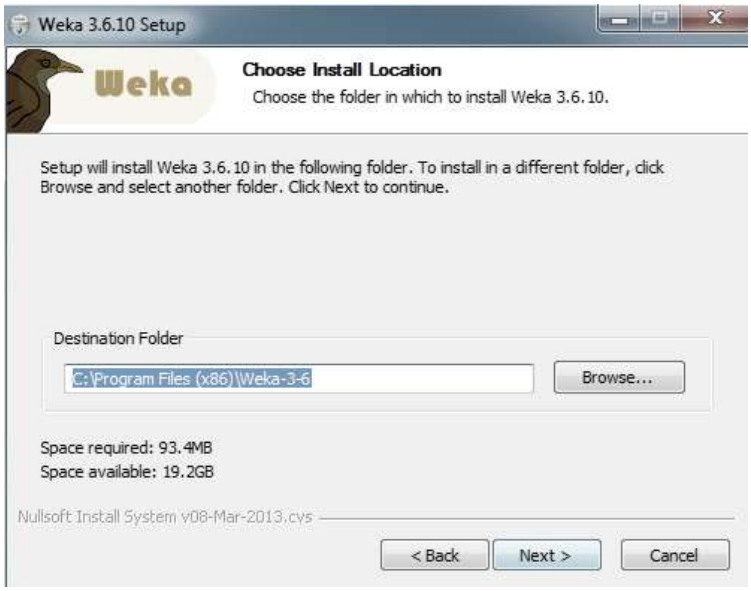
2. Hacer clic en la opción *Next* para iniciar la instalación.

3. Hacer clic en la opción *I Agree* para aceptar los términos de instalación.



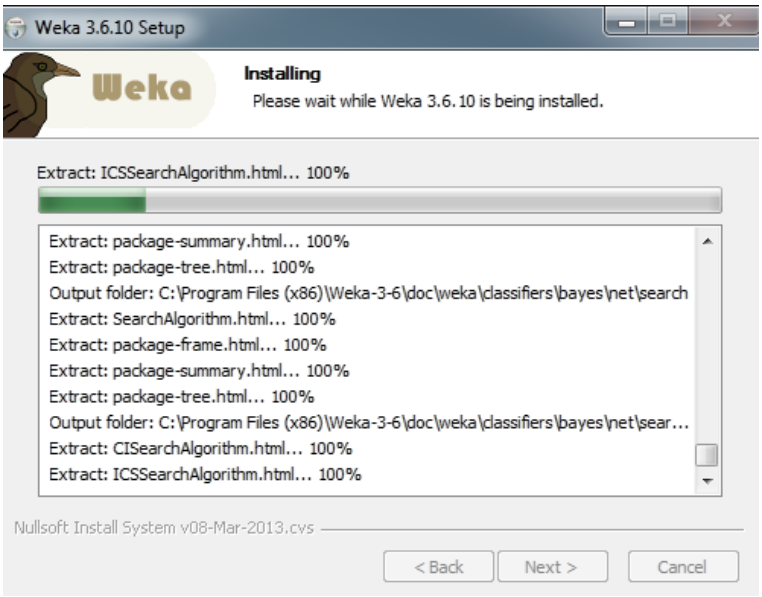
4. Hacer clic en la opción *Next* para continuar con la instalación.

5. Hacer clic en la opción *Next* para continuar con la instalación.



6. Hacer clic en la opción *Install* para continuar con la instalación.

7. Después de realizar la opción *Install* podrá observar la siguiente ventana, esta indica el porcentaje de instalación.

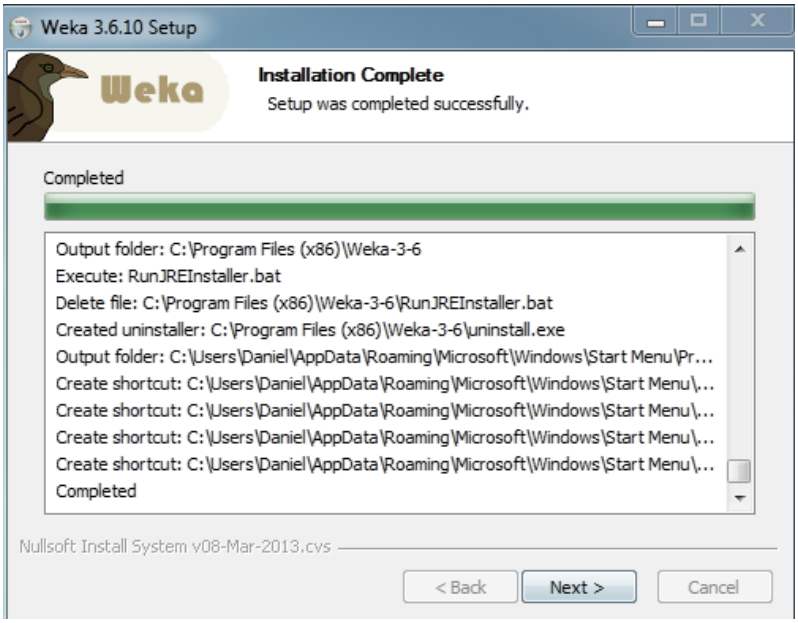


8. Hacer clic en la opción *Instalar* para realizar la instalación de los complementos de Java.

9. La siguiente ventana muestra el proceso de configuración de Java.

10. Hacer clic en la opción *Cerrar* para terminar con la configuración de Java.

11. La siguiente venta visualiza el porcentaje de instalación de la herramienta Weka.



12. Hacer clic en la opción *Finish* para concluir con la instalación.

13. Finalmente, una vez realizada la instalación en su totalidad, la siguiente ventana se desplegará. Esto indica que el proceso de instalación se completó con éxito.

Bibliografía

[1] Advances in knowledge discovery. Fayyad usama m. Pearson (1996)

[2] Introducción a la minería de datos. Hernández orallo José. Pearson (2004)

[3] Inteligencia artificial un enfoque moderno. Stuart Russell. Alfa omega (2004).

[4] Inteligencia artificial. Pedro Ponce cruz. Alfa omega (2010).

[5] Inteligencia artificial técnicas, métodos y aplicaciones. José t. Palma Méndez, Roque Martin Morales. Alfa omega (2008).

[6] Essentials of artificial inteligence. Matt Ginsberg. Morgan kaufman (1993)

[7] Machine Learning Group at the University of Waikato. Weka 3: Data Mining Software in Java; [Citado 2014 Enero 3] Disponible en: http://www.cs.waikato.ac.nz/ml/weka/

[8] Ian H. Witten, Eibe Frank, Mark A. Hall. Data Mining Practical Machine Learning Tools and Techniques. 3ra Ed; 2011. Páginas [403 - 585].

PRÁCTICA 2

ENTORNO DE TRABAJO DEL SOFTWARE WEKA

Objetivo

Introducir al alumno al entorno de trabajo del software WEKA.

Introducción

El Aprendizaje Automatizado (AA) es una rama de la Inteligencia Artificial (IA), cuyo objetivo es desarrollar técnicas que permitan crear programas que puedan aprender de forma similar a lo realizado por los humanos, es decir, aprender por sí mismos; capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del mismo se solapa con el de la estadística, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automatizado se centra más en el estudio de la complejidad computacional de los problemas.

Existen varias herramientas o plataformas para la utilización de los métodos de aprendizaje automatizado y con ellos el desarrollo de modelos para la solución de diversos problemas, como es el caso de Weka, que constituye un entorno de trabajo el cual integra una amplia colección de algoritmos. Desarrollado en lenguaje Java por un equipo de investigadores de la universidad de Waikato (Nueva Zelanda), bajo licencia GNU (General Public License), y se caracteriza por la independencia de su arquitectura, ya que funciona en cualquier plataforma sobre la que existe una máquina virtual Java disponible. Permite aplicar, analizar y evaluar algunas de las técnicas más relevantes del análisis de datos, dentro de las que se enmarcan: el pre-procesamiento de datos, clasificación, regresión, agrupamiento, reglas de asociación y visualización. Es un software libre que está orientado a la extensibilidad, por lo que es posible añadir nuevas funcionalidades. Todo lo anterior justifica que sea una de las herramientas más utilizadas en la minería de datos.

Desarrollado bajo licencia GPL lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años, incluye las siguientes características: Diversas fuentes de datos (ASCII, JDBC), Interfaz visual basado en procesos/flujos de datos (rutas), Distintas herramientas de minería de datos: reglas de asociación (a priori, Tertius,), agrupación/segmentación/conglomerado (Cobweb, EM y k-medias), clasificación (redes neuronales, reglas y árboles de decisión, aprendizaje Bayesiona) y regresión (Regresión lineal, SVM..), Manipulación de datos (pick & mix, muestreo, combinación y separación), combinación de modelos (Bagging, Boosting ...), Visualización anterior (datos en múltiples gráficas) y posterior (árboles, curvas ROC, curvas de coste..) y entorno de experimentos, con la posibilidad de realizar pruebas estadísticas (t-test).

Desarrollo

El alumno ejecutara paso a paso lo que a continuación se describe. Una vez que Weka esté en ejecución aparecerá una ventana denominada selector de interfaces, que nos permite seleccionar la interfaz con la que se desea comenzar a trabajar. Las posibles interfaces a seleccionar son: *Simple CLI, Explorer, Experimenter y Knowledge flow*.(véase Figura 1.).

Figura 1. Selector de interfaces de WEKA

EXPLORER: es la opción que permite llevar a cabo la ejecución de los algoritmos de análisis implementados sobre los archivos de entrada, una ejecución independiente por cada prueba. *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 (véase Figura 2): **Preprocess:** selección de la fuente de datos y preparación (filtrado), **Clasify:** 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.

Figura 2. Opción Explorer

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** (Figura 3) 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.

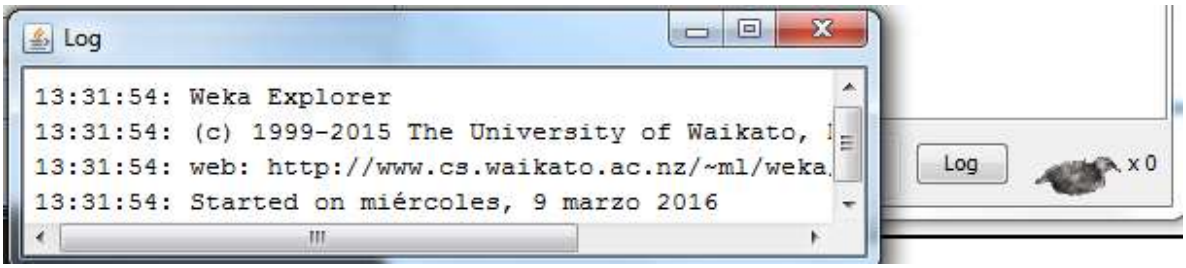


Figura 3. Botón *log* de Explorer.

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 (Figura 4).

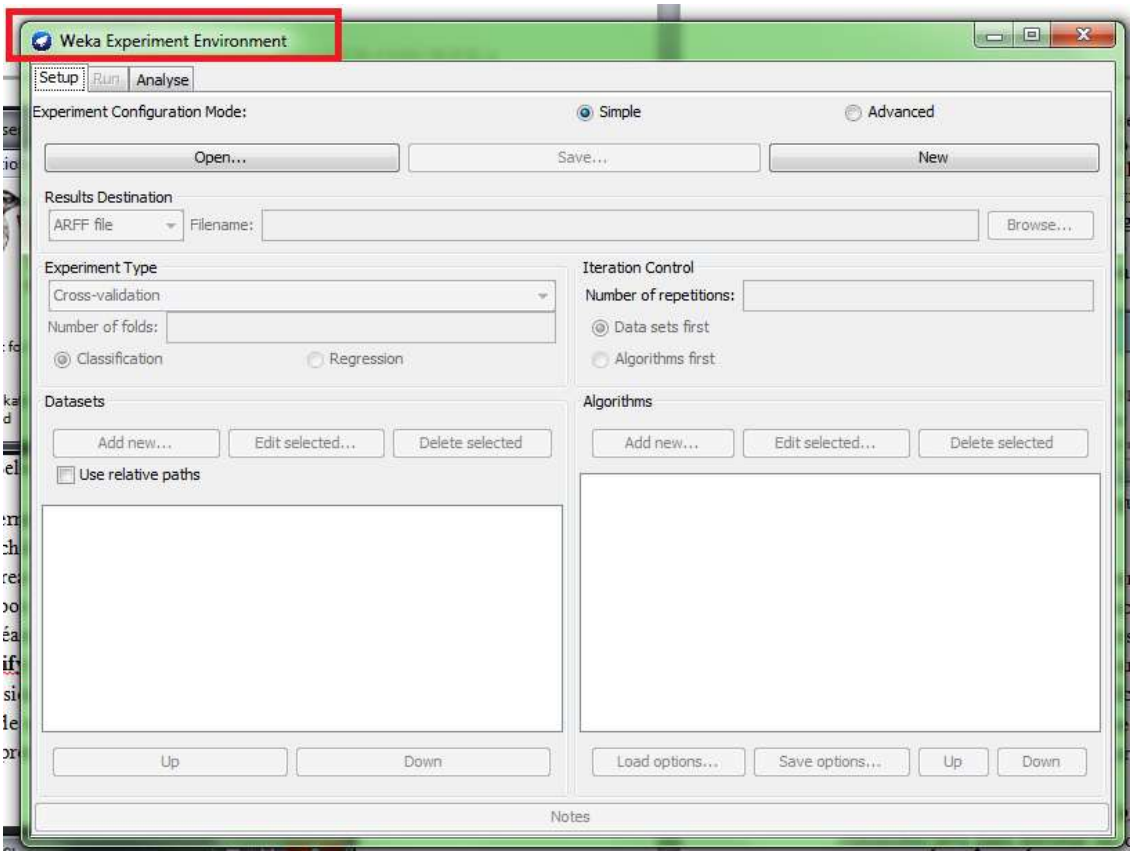


Figura 4. Opción Experimenter

KnowledgeFlow: 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 (Figura 5).

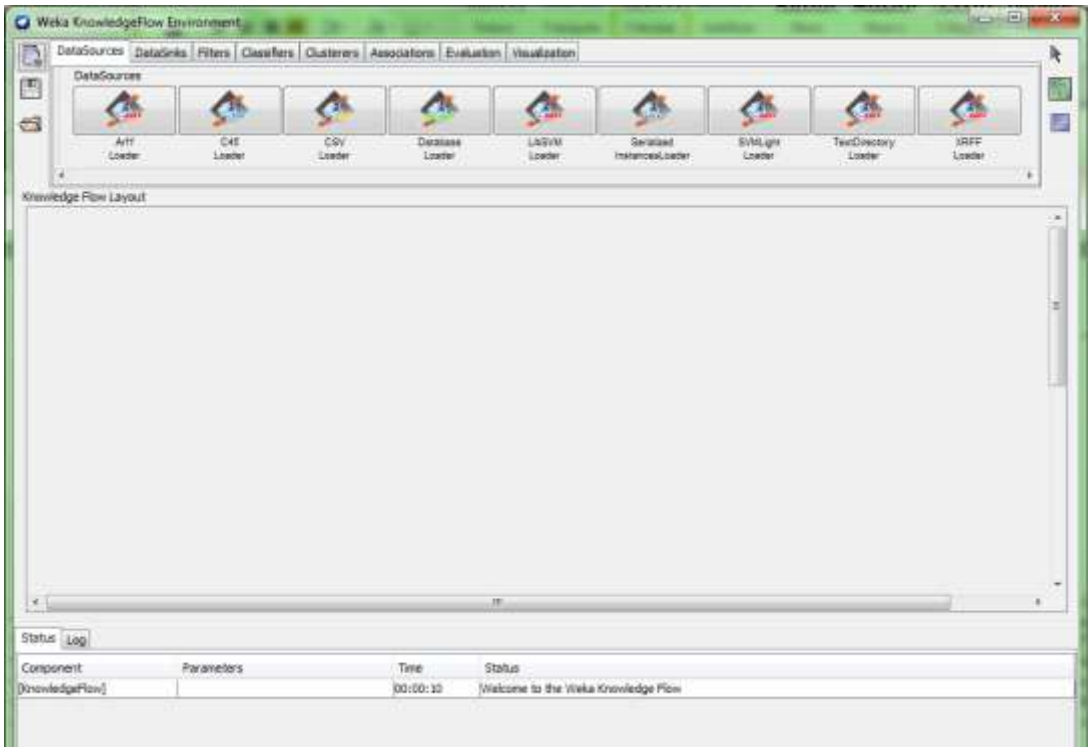


Figura 5. Opción KnowledgeFlow

Simple CLI: la interfaz "Command-Line Interfaz" es simplemente una ventana de comandos java para ejecutar las clases de WEKA.

Conclusiones
Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Acervo bibliográfico
Básico:

1. Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones,
ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
2. Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
3. Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
4. Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

-

Bibliografía Complementaria

1. An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
2. - Practical Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
3.- Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
4.- Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.

PRÁCTICA 3

PREPARACIÓN DE LOS DATOS

Objetivo
El alumno aprenderá a codificar los datos en el formato específico de WEKA.

Introducción

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.

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.

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: archivo 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 archivos de texto. Los datos deben estar dispuestos en el archivo de la forma siguiente: cada instancia en una fila, y con los atributos separados por comas. El formato de un archivo arff tiene 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
```

Desarrollo

El alumno generara un archivo de Weka que contenga los datos de un alumno, su carrera así como las materias que toma, deberá especificar los nombres de cada columna precedido por @Attribute y al final el tipo de archivo, en este caso, para fines prácticos, Justo antes de los datos, poner @Data, Guardar el archivo como ANSI y con la extensión .arff, suba su archivo y capturas de pantalla de la realización y carga del archivo.
Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones
Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía

Básico:
Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España
Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.
Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria
An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press
Practical Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley
Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.
Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.

PRÁCTICA 4

TRABAJO CON FILTROS: PREPARACIÓN DE ARCHIVOS DE MUESTRA

Objetivo
El alumno utilizara los filtros para manipular los datos que requiera.

Introducción

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 (Figura 1). 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.

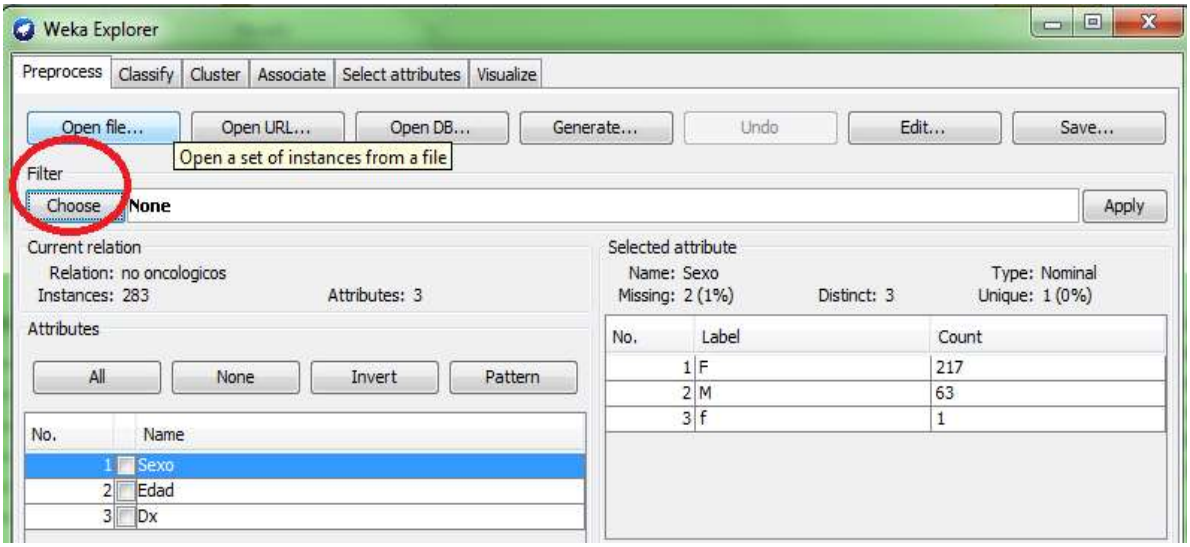


Figura 1. Filtros aplicables en WEKA

Además, pueden guardarse los resultados de aplicar filtros en nuevos archivos, 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 (Figura 2).

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 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.

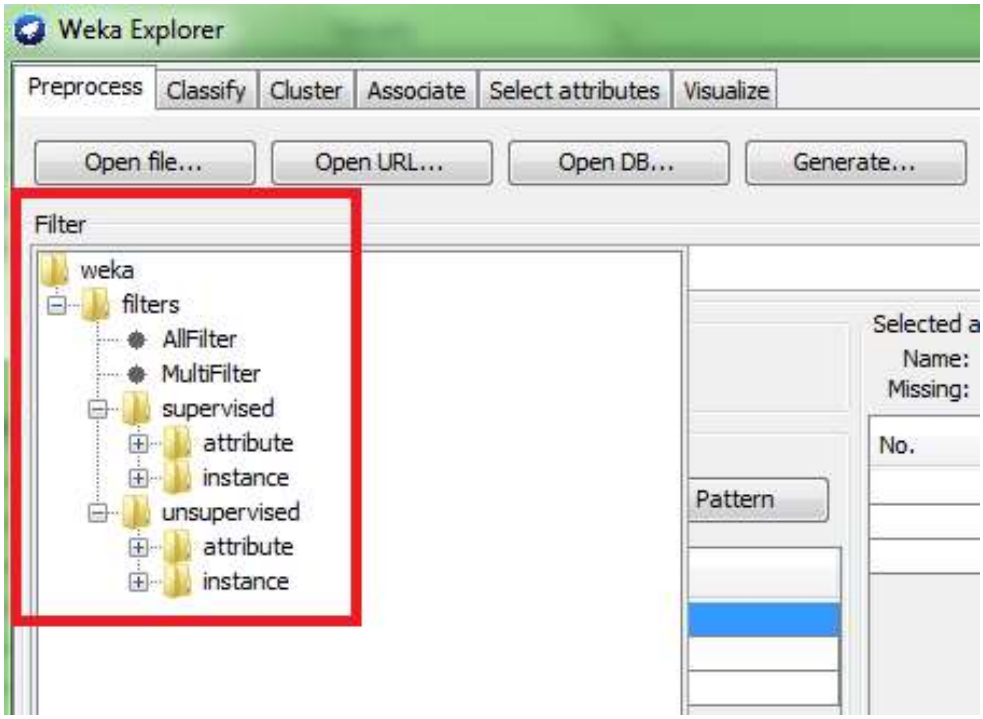


Figura 2. Filtros supervisados y no supervisados.

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, indicándolo en el cuadro de configuración del filtro *Remove* (Figura 3).

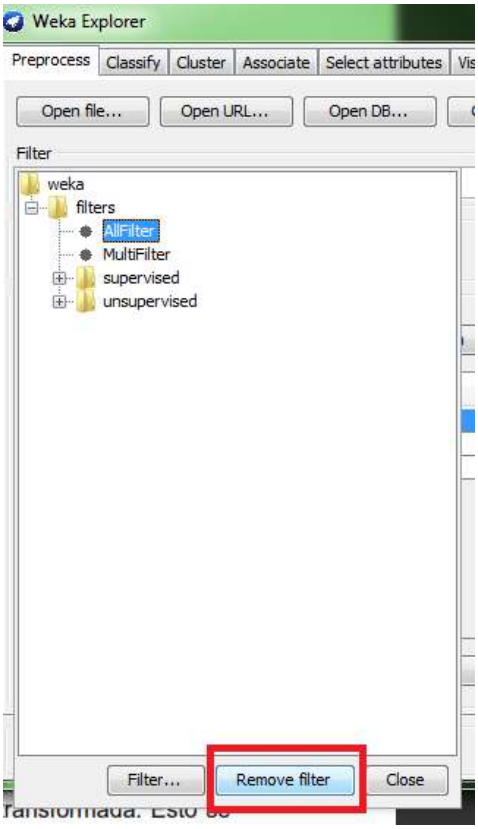


Figura 3. Remover atributos

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 boton **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 archivo ARFF con el botón **Save**, dentro de la ventana **Preprocess** (Figura 4).

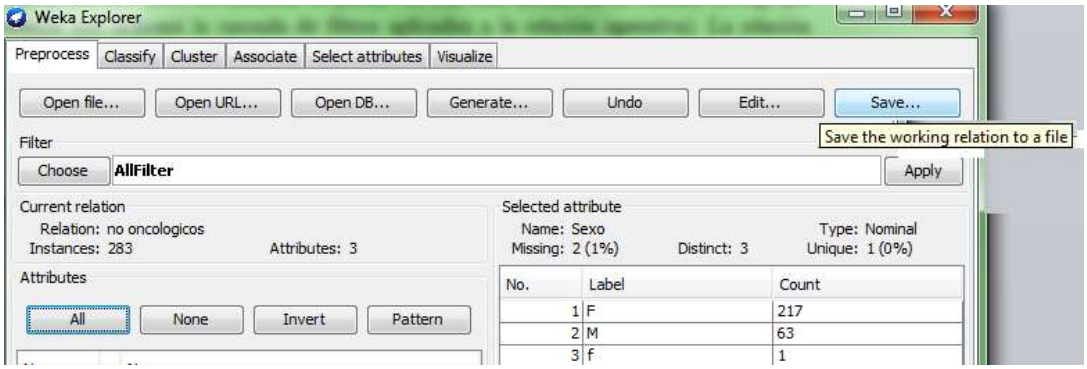


Figura 4. Guardando cambios.

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. 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 (Figura 5).

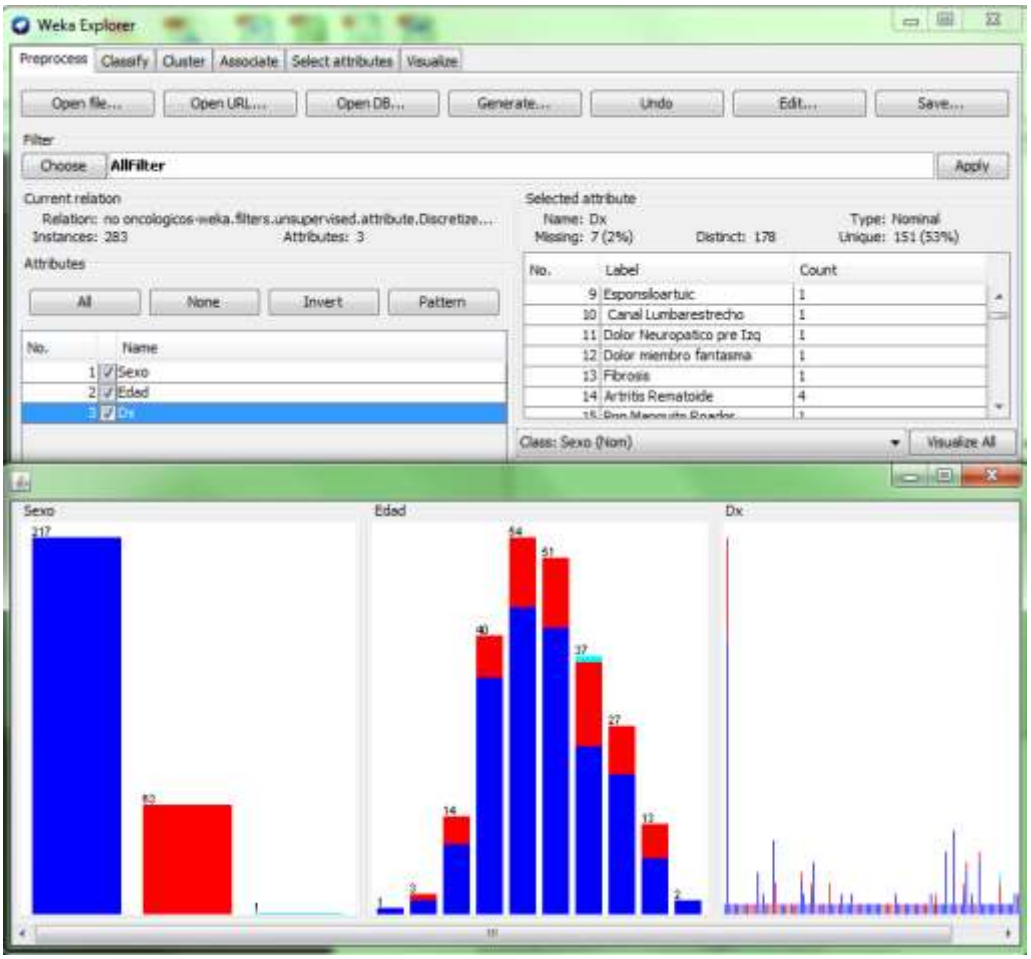


Figura 5. Filtro de discretización.

Filtros de añadir expresiones

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 (Figura 6).

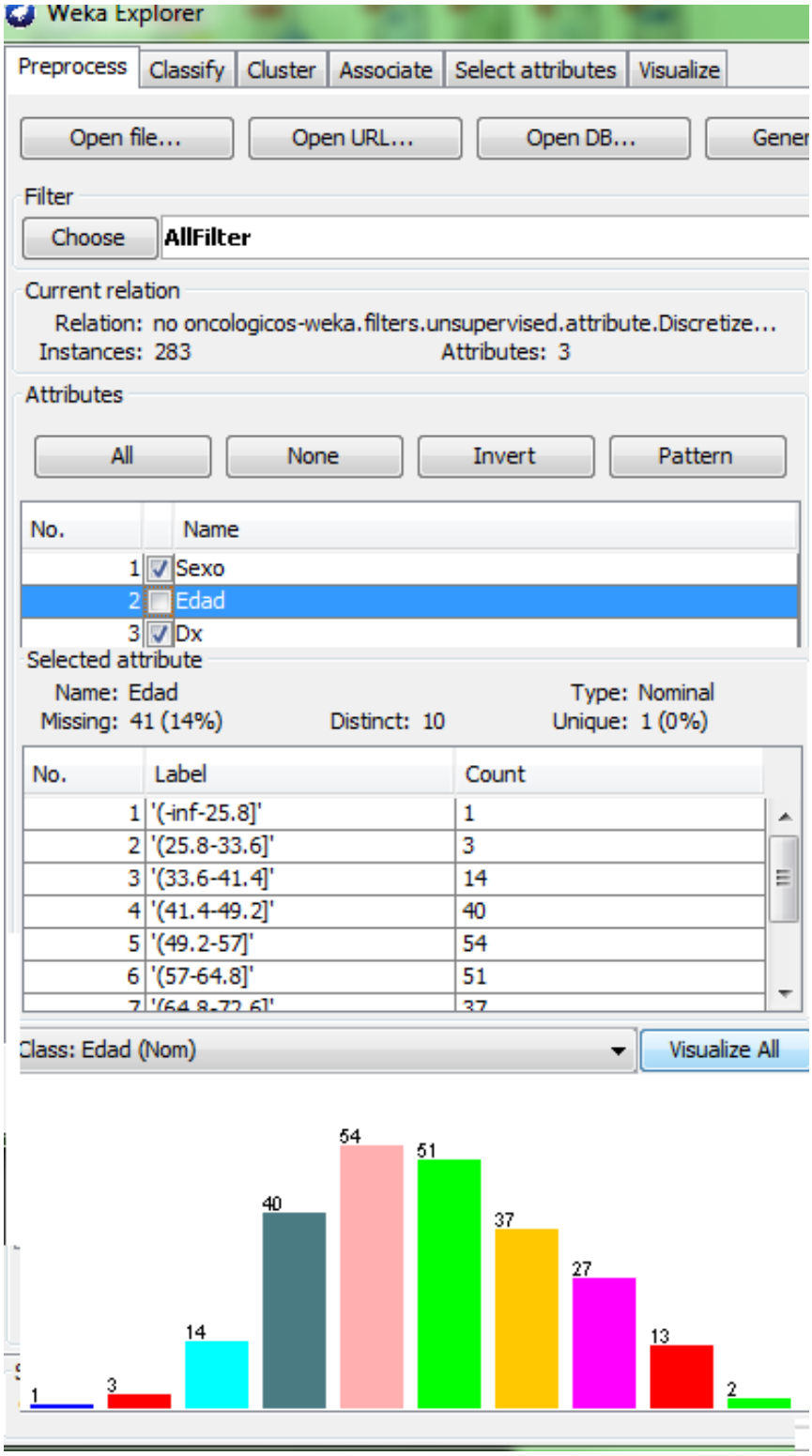


Figura 6. Filtros de añadir expresiones

Desarrollo

El alumno aplicara los filtros anteriormente expuestos, del archivo proporcionado en clase, explique lo que observa, documente y adjunte a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía

Básico:

Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España

Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.

Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press

Practical Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley

Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.

Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.

PRÁCTICA 5

PROCESAMIENTO DE DATOS

Objetivo
El alumno determinara la relación entre atributos.

Introducción

La búsqueda de patrones secuenciales se enfrenta a características de los datos que pueden perjudicar los resultados de una investigación. Algunas de estas características son la presencia de ruido en los datos y la existencia masiva de patrones que no son de interés para el caso de estudio. Estas situaciones deben ser consideradas, especialmente en las fases de preprocesamiento de los datos y luego de realizada la búsqueda de patrones.

Los datos reales están "sucios" y pueden presentar alguna de estas características: Datos incompletos: valores de atributos inexistentes, Datos con ruido: errores de precisión, errores de medición, errores de almacenamiento. Datos inconsistentes: outliers.

Para obtener conclusiones válidas y útiles al aplicar minería de datos, es necesaria una adecuada preparación de los datos previa al proceso de minería, la gran variedad de métodos estadísticos y computacionales para investigar la existencia de relaciones y patrones de comportamiento en datos almacenados electrónicamente. Estas relaciones o patrones emergentes pueden sugerir explicaciones causales que puedan ser verificadas posteriormente o bien pueden sugerir estrategias de acción para lograr ciertos objetivos de cambio.

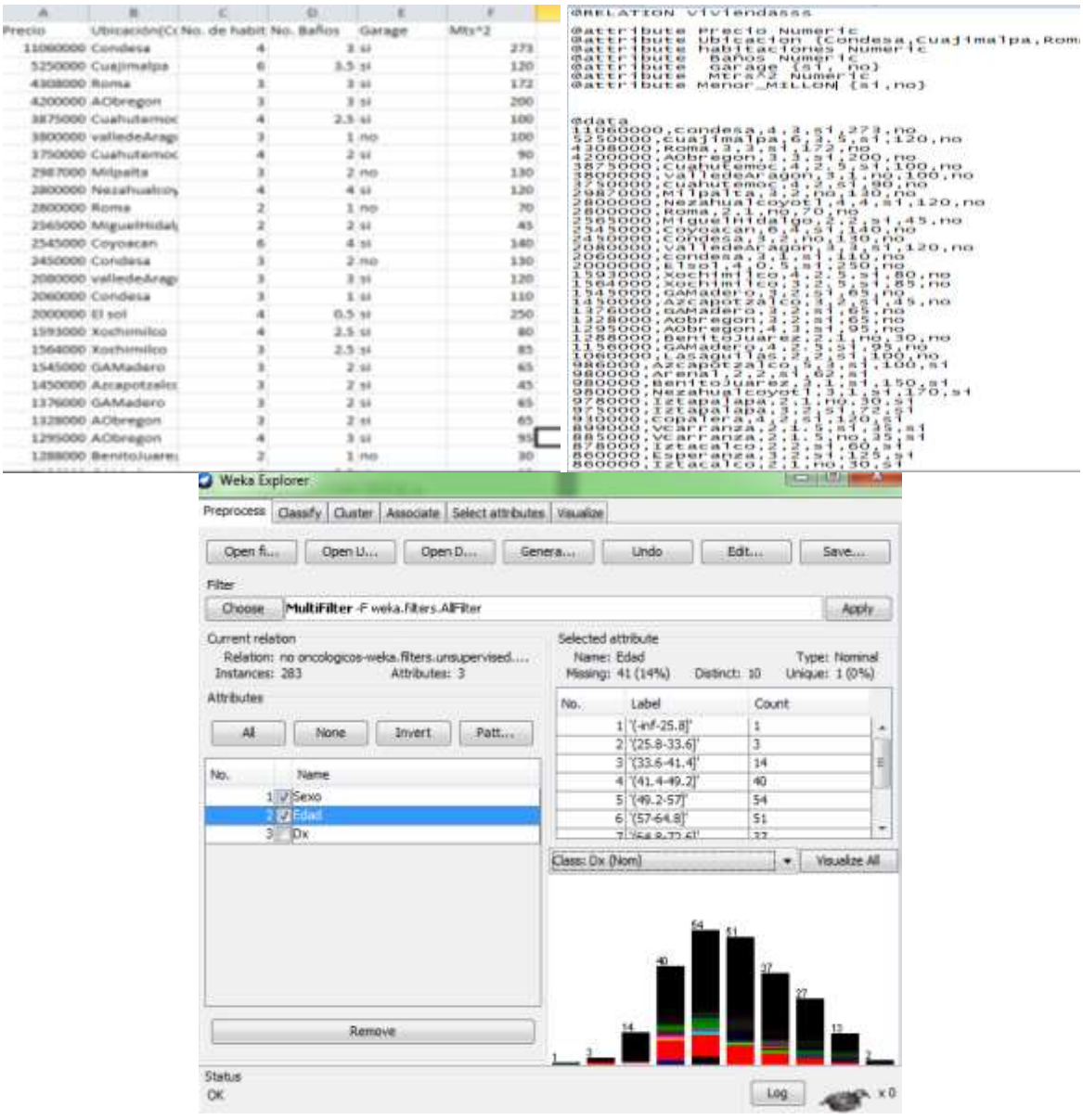
Las etapas de preprocesamiento y codificación son abarcadas simultáneamente. Esto se logra mediante el uso de una de las herramientas implementadas en esta investigación, llamada "Preprocesador Logs", junto con las opciones de pre procesamiento que entrega Weka, y en menor medida en modificaciones manuales que son necesarias sobre determinados atributos.

El principal objetivo que se persigue en las etapas de preprocesamiento y codificación, es lograr modificar los datos para un óptimo trabajo con los algoritmos de Minería de Datos. Esto se debe realizar siempre cuidando no perder ni alterar el significado y las propiedades de los datos analizados. Buscando este objetivo, se realizan una serie de modificaciones a los datos. La primera modificación aplicada es la estandarización de los ejecutores y tareas que se presentan en los procesos. Dada la variabilidad, y en algunos casos la extensa denominación que presentan estos atributos, se implementó una serie de instrucciones para poder dar un nombre estándar a cada tarea y cada ejecutor.

Desarrollo

1. Abra Weka y seleccione Explorer. Lo primero que vamos a hacer es cargar los datos en el área de trabajo. Para ello, pincha en el botón "Open file" del entorno "preprocess". Seleccionamos el archivo "tasación.arff" y si todo ha ido bien veremos la pantalla, Weka utiliza un formato específico de datos, el formato arff. Un archivo con este formato, no solo contiene los datos desde donde vamos a efectuar el aprendizaje, además incluye meta información sobre los propios datos, como por ejemplo el nombre y tipo de cada atributo, así como una descripción textual del origen de los datos.

Podemos convertir archivos en texto conteniendo un registro por línea y con los atributos separados con comas (formato csv) a archivo arff mediante el uso de un filtro convertidor.



Si seleccionamos cada uno de los atributos, conoceremos más información del atributo en cuestión: tipo (nominal o numérico), valores distintos, registros que no tienen información de ese atributo, el valor máximo y mínimo (solo en atributos numéricos), y finalmente un histograma con información sobre la distribución de los ejemplos para ese atributo, reflejando con el uso de colores la distribución de clases de cada uno de los registros.

Se aplicara el algoritmo J48 de WEKA es una implementación del algoritmo C4.5, uno de los algoritmos de minería de datos más utilizado. El parámetro más importante que deberemos tener en cuenta es el factor de confianza para la poda "confidence level", que influye en el tamaño y capacidad de predicción del árbol construido.

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     Modificadoviviendass2
Instances:    40
Attributes:   6
              Precio
              Ubicación(Col. O Del)
              No. de habitaciones
              No. Baños
              Garage
              Mts^2
Test mode:evaluate on training data

=== Classifier model (full training set) ===
```

Muestra los atributos que son clasificados.

```
Precio <= 1545000
|   Precio <= 978000
|   |   Precio <= 878000
|   |   |   Precio <= 783000: Vicente Guerrero (2.0)
|   |   |   Precio > 783000: Iztacalco (3.0/1.0)
|   |   Precio > 878000
|   |   |   Precio <= 899000: VCarranza (2.0)
|   |   |   Precio > 899000: Iztapalapa (3.0/1.0)
|   Precio > 978000
|   |   No. Baños <= 1.5: BenitoJuarez (3.0/1.0)
|   |   No. Baños > 1.5
|   |   |   No. de habitaciones <= 2: Las aguilas (2.0/1.0)
|   |   |   No. de habitaciones > 2
|   |   |   |   No. Baños <= 2.5: GAMadero (5.0/2.0)
|   |   |   |   No. Baños > 2.5: AObregon (2.0/1.0)
Precio > 1545000
|   No. de habitaciones <= 3
|   |   No. de habitaciones <= 2: Roma (2.0/1.0)
|   |   No. de habitaciones > 2
|   |   |   Mts^2 <= 150
|   |   |   |   No. Baños <= 2
|   |   |   |   |   Precio <= 2565000: Condesa (2.0)
|   |   |   |   |   Precio > 2565000: valledeAragon (2.0/1.0)
|   |   |   |   No. Baños > 2: valledeAragon (2.0/1.0)
|   |   |   Mts^2 > 150: Roma (2.0/1.0)
|   No. de habitaciones > 3
|   |   No. de habitaciones <= 5
|   |   |   No. Baños <= 2.5
|   |   |   |   Precio <= 2800000: El sol (2.0/1.0)
|   |   |   |   Precio > 2800000: Cuahutemoc (2.0)
|   |   |   No. Baños > 2.5: Condesa (2.0/1.0)
|   |   No. de habitaciones > 5: Cuajimalpa (2.0/1.0)

Number of Leaves  :     17

Size of the tree :      33

Time taken to build model: 0.01 seconds
```

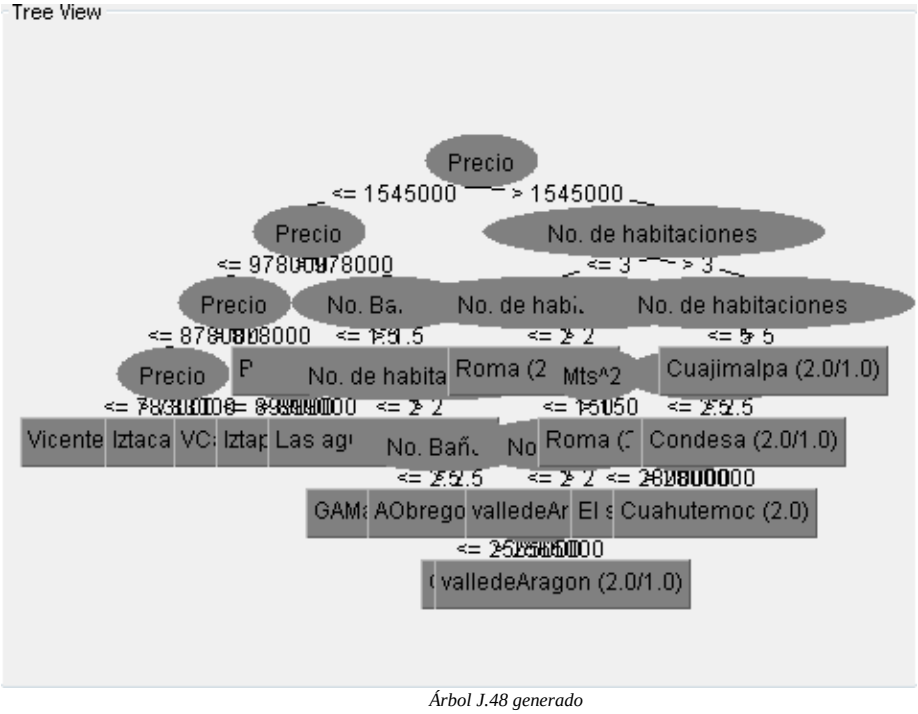
Pseudocódigo del árbol.

Esta clasificación nos permite perfectamente visualizar a través de diversas condiciones cual es la vivienda que conveniente con relación a su precio y con otros atributos.

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          26               65       %
Incorrectly Classified Instances        14               35       %
Kappa statistic                          0.6304
Mean absolute error                      0.0343
Root mean squared error                  0.131
Relative absolute error                 41.475  %
Root relative squared error             64.4523 %
Total Number of Instances               40
```

Resultados de la clasificación

El error de clasificación que se produce es de 35% y solo se clasifican 26 instancias lo cual lo convierte en un algoritmo no muy óptimo para este caso. Pero si se encuentra entre los mejores.



Árbol J.48 generado

Para el árbol de decisión creado el valor de confianza es del 25% Según baje este valor, se permiten más operaciones de poda. El valor más importante que nos permite visualizar la clasificación correctamente es el atributo de Precio y para posteriormente clasificar las demás características.

Aplicando Random Tree

Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.
Resulta ser u algoritmo muy optimo a la hora de realizar la clasificación

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          40               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               40
```

Resultados de la clasificación

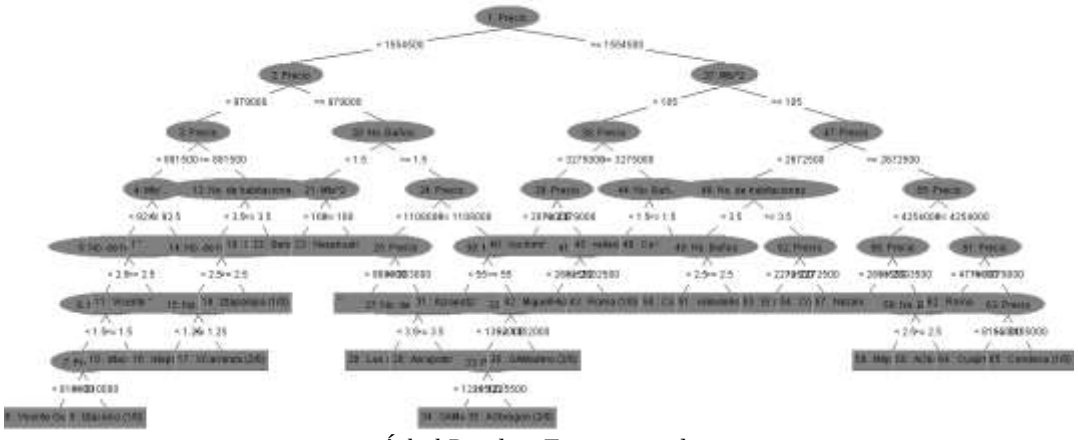
```
=== Confusion Matrix ===

 a b c d e f g h i j k l m n o p q r s t u v w   <-- classified as
 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = Condesa
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | b = Cuajimalpa
 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = Roma
 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = AObregon
 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | e = Cuahutemoc
 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | f = valledeAragon
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = Elsol
 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = Xochimilco
 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = GAMadero
 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 | j = Azcapotzalco
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 | k = Coyoacan
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 | l = BenitoJuarez
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 | m = Lasaguilas
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 | n = Arenal
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 | o = Nezahualcoyotl
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 | p = Iztapalapa
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 | q = Copalera
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 | r = VCarranza
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 | s = Iztacalco
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 | t = Esperanza
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 | u = VicenteGuerrero
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 | v = MiguelHidalgo
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 | w = Milpalta
```

Matriz de Congruencia

No existen elementos nulos fuera de la matriz principal. Lo que lo convierte en altamente confiable.

```
RandomTree
==========

Precio < 1554500
|   Precio < 979000
|   |   Precio < 881500
|   |   |   Baños < 1.5
|   |   |   |   Garage = si : VicenteGuerrero (2/0)
|   |   |   |   Garage = no : Ixtacalco (1/0)
|   |   |   Baños >= 1.5
|   |   |   |   habitaciones < 2.5 : Ixtacalco (1/0)
|   |   |   |   habitaciones >= 2.5 : Esperanza (1/0)
|   |   Precio >= 881500
|   |   |   Mtrs^2 < 96
|   |   |   |   Baños < 1.25 : Iztapalapa (1/0)
|   |   |   |   Baños >= 1.25
|   |   |   |   |   habitaciones < 2.5 : VCarranza (2/0)
|   |   |   |   |   habitaciones >= 2.5 : Iztapalapa (1/0)
|   |   |   Mtrs^2 >= 96 : Copalera (1/0)
|   Precio >= 979000
|   |   habitaciones < 2.5
|   |   |   Baños < 1.5 : BenitoJuarez (1/0)
|   |   |   Baños >= 1.5
|   |   |   |   Mtrs^2 < 81 : Arenal (1/0)
|   |   |   |   Mtrs^2 >= 81 : Lasaguilas (1/0)
|   |   habitaciones >= 2.5
|   |   |   Baños < 1.5
|   |   |   |   Mtrs^2 < 160 : BenitoJuarez (1/0)
|   |   |   |   Mtrs^2 >= 160 : Nezahualcoyotl (1/0)
|   |   |   Baños >= 1.5
|   |   |   |   Precio < 1352000
|   |   |   |   |   habitaciones < 4.5
|   |   |   |   |   |   habitaciones < 3.5 : ADbregon (1/0)
|   |   |   |   |   |   habitaciones >= 3.5
|   |   |   |   |   |   |   Precio < 1225500 : GAMadero (1/0)
|   |   |   |   |   |   |   Precio >= 1225500 : ADbregon (1/0)
|   |   |   |   |   habitaciones >= 4.5 : Azcapotzalco (1/0)
|   |   |   |   Precio >= 1352000
|   |   |   |   |   Mtrs^2 < 55 : Azcapotzalco (1/0)
|   |   |   |   |   Mtrs^2 >= 55 : GAMadero (2/0)
Precio >= 1554500
|   Mtrs^2 < 105
|   |   habitaciones < 2.5
|   |   |   Baños < 1.5 : Roma (1/0)
|   |   |   Baños >= 1.5 : MiguelHidalgo (1/0)
|   |   habitaciones >= 2.5
|   |   |   Mtrs^2 < 87.5 : Xochimilco (2/0)
|   |   |   Mtrs^2 >= 87.5
|   |   |   |   habitaciones < 3.5 : valledeAragon (1/0)
|   |   |   |   habitaciones >= 3.5 : Cuahutemoc (2/0)
|   Mtrs^2 >= 105
|   |   Precio < 2672500
|   |   |   Baños < 2.5
|   |   |   |   Baños < 0.75 : Elsol (1/0)
|   |   |   |   Baños >= 0.75 : Condesa (2/0)
|   |   |   Baños >= 2.5
|   |   |   |   Precio < 2312500 : valledeAragon (1/0)
|   |   |   |   Precio >= 2312500 : Coyoacan (1/0)
|   |   Precio >= 2672500
|   |   |   Precio < 4256000
|   |   |   |   Precio < 2593500 : Nezahualcoyotl (1/0)
|   |   |   |   Precio >= 2593500
|   |   |   |   |   Baños < 2.5 : Milpalta (1/0)
|   |   |   |   |   Baños >= 2.5 : ADbregon (1/0)
|   |   |   Precio >= 4256000
|   |   |   |   Precio < 4779000 : Roma (1/0)
|   |   |   |   Precio >= 4779000
|   |   |   |   |   Precio < 8155000 : Cuajimalpa (1/0)
|   |   |   |   |   Precio >= 8155000 : Condesa (1/0)

Size of the tree : 67
```

Como puede observarse en el algoritmo hace una máxima clasificación de los datos. Haciendo como en el árbol anterior J-48 una comparación total discerniendo entre cada uno de los datos.



Árbol Random Tree generado

Muestra un árbol más complejo que el anterior que era de tamaño 33 y ahora de tamaño 67 con todas las instancias clasificadas. El atributo más importante en esta clasificación es el valor del precio. Cada rama del árbol va ir desglosándose hasta obtener sus últimas hojas.

Aplicando NaiveBayes

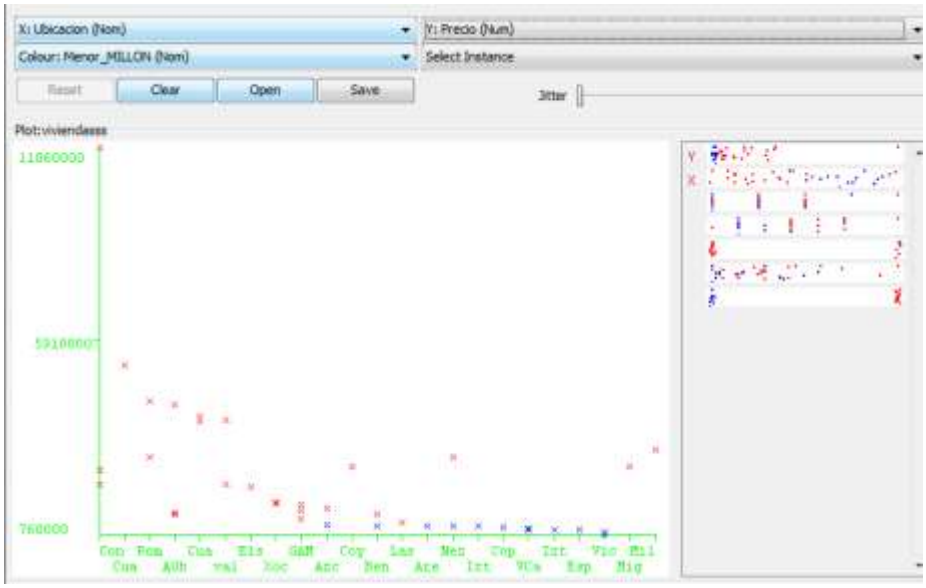
```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          30               75      %
Incorrectly Classified Instances        10               25      %
Kappa statistic                          0.737
Mean absolute error                      0.024
Root mean squared error                  0.1267
Relative absolute error                 29.0318 %
Root relative squared error             62.3044 %
Total Number of Instances               40

=== Detailed Accuracy By Class ===

=== Confusion Matrix ===

 a b c d e f g h i j k l m n o p q r s t u v w   <-- classified as
 1 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = Condesa
 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | b = Cuajimalpa
 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = Roma
 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = AObregon
 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | e = Cuahutemoc
 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | f = valledeAragon
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = Elsol
 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = Xochimilco
 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = GAMadero
 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 | j = Azcapotzalco
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 | k = Coyoacan
 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 | l = BenitoJuarez
 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 | m = Lasaguilas
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 | n = Arenal
 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 | o = Nezahualcoyotl
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 | p = Iztapalapa
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 | q = Copalera
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 | r = VCarranza
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 | s = Iztacalco
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 | t = Esperanza
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 | u = VicenteGuerrero
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 | v = MiguelHidalgo
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 | w = Milpalta
```

Matriz de confusión

```
=== Classifier model (full training set) ===

Naive Bayes Classifier

                        Class
Attribute               Condesa      Cuajimalpa          Roma       AObregon
                         (0.06)         (0.03)         (0.05)         (0.06)
================================================================================
Precio
  mean              5199047.619    5297142.8571   3678571.4286   2256190.4762
  std. dev.       4232897.6154      49047.619    735714.2857   1323377.1133
  weight sum                 3              1              2              3
  precision         294285.7143    294285.7143    294285.7143    294285.7143

habitaciones
  mean                   3.3333              6            2.5         3.3333
  std. dev.              0.4714         0.1667            0.5         0.4714
  weight sum                  3              1              2              3
  precision                   1              1              1              1

Baños
  mean                        2            3.5              2         2.6667
  std. dev.              0.8165         0.0833              1         0.4714
  weight sum                  3              1              2              3
  precision                 0.5            0.5            0.5            0.5

Garage
  si                        3.0            2.0            2.0            4.0
  no                        2.0            1.0            2.0            1.0
  [total]                   5.0            3.0            4.0            5.0


    GAMadero     Azcapotzalco        Coyoacan     BenitoJuarez
      (0.06)         (0.05)          (0.03)         (0.05)
================================================================

1373333.3333   1177142.8571   2648571.4286        1030000
 138727.6161    294285.7143     49047.619     147142.8571
           3              2              1              2
 294285.7143    294285.7143    294285.7143    294285.7143


       3.3333              4              6            2.5
       0.4714              1         0.1667            0.5
            3              2              1              2
            1              1              1              1


       2.1667            2.5              4              1
       0.2357            0.5         0.0833         0.0833
            3              2              1              2
          0.5            0.5            0.5            0.5


          4.0            3.0            2.0            2.0
          1.0            1.0            1.0            2.0
          5.0            4.0            3.0            4.0


       70.875         70.875         141.75         91.125
      14.3189         30.375         1.6875          60.75
            3              2              1              2
      10.125         10.125         10.125         10.125
```

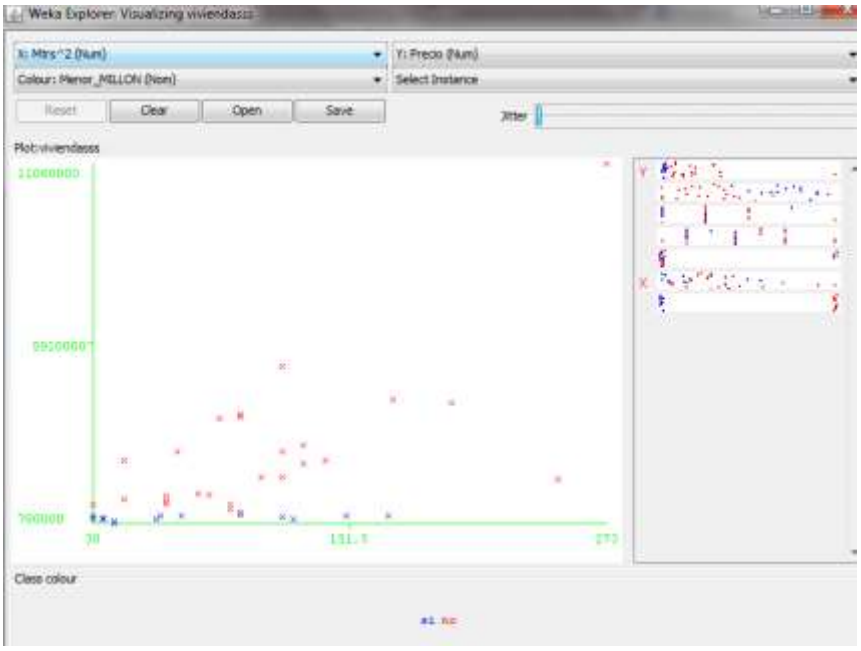
Resultados de la Clasificación

Muestra una serie de columnas con las respectivas ubicaciones y sus datos correspondientes como desviación estándar, media, valores máximos y valores mínimos y su clasificación con respecto a los demás atributos.



Visualización Ubicación-Precio

Muestra la relación que existe entre la ubicación de las 40 viviendas y su precio.



Visualización de M^2 y Precio

Se puede visualizar las instancias en una grafico mostrando la relación que existe entre el tamaño de vivienda y el precio.

Clasificador	Instancias clasificadas correctamente	%
Random Tree	40	100
IB1	40	100
NaiveBayes	30	75
J-48	26	65
PART	24	60
OneR	4	10

Tabla 1Resultados de la clasificación

Después de diversas pruebas utilizando los clasificadores se puede obtener la tabla anterior mostrando con mayor eficiencia en este caso al Random Tree.

Clasificador	Tasa de Error	Rapidez	Interpretabilidad	Simplicidad
Random Tree	0%	0.01 s	Arbol	Buena
IB1	0%	0 s	Lazy	Buena
NaiveBayes	25%	0s	Bayes	Buena
J-48	35%	0.01	Arbol	Regular
PART	40%	0.01s	Reglas	Regular
OneR	90%	0 s	Reglas	Malo

Criterios de Validación del Clasificador

Estimar la bondad del clasificador sirve para medir la capacidad de predicción del clasificador. Como se puede observar la tasa de error nos da una idea del porcentaje de éxito en el proceso. La rapidez nos indica que tan rápido puede construir el modelo clasificado.

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía

Básico:

Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España

Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.

Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press

Practical Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley

Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.

Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.

PRÁCTICA 6

AGRUPAMIENTO NUMÉRICO

Objetivo
El alumno conocerá el concepto de agrupamiento y lo aplicara en Weka

Introducción

Los algoritmos de agrupamiento buscan grupos de instancias con características similares, según un criterio de comparación entre valores de atributos de las instancias definidos en los algoritmos.

Agrupamiento Numérico

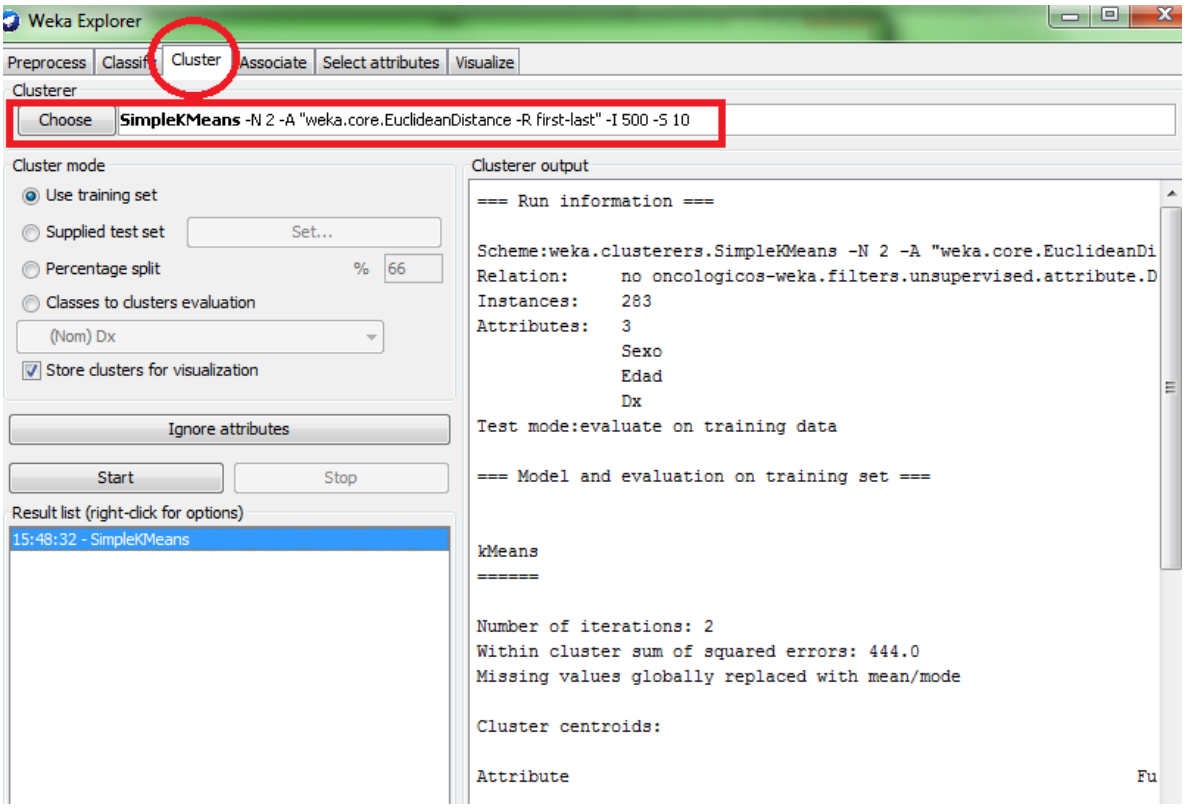
Algoritmo K-Medias: Se trata de un algoritmo clasificado como Método de Particionado y Recolocación. Este método es hasta ahora el más utilizado en aplicaciones científicas e industriales. El nombre le viene porque representa cada uno de los clúster por la media (o media ponderada) de sus puntos, es decir, por su centroide. Este método únicamente se puede aplicar a atributos numéricos, y los *outliers* le pueden afectar muy negativamente. Sin embargo, la representación mediante centroides tiene la ventaja de que tiene un significado gráfico y estadístico inmediato. La suma de las discrepancias entre un punto y su centroide, expresado a través de la distancia apropiada, se usa como función objetivo.

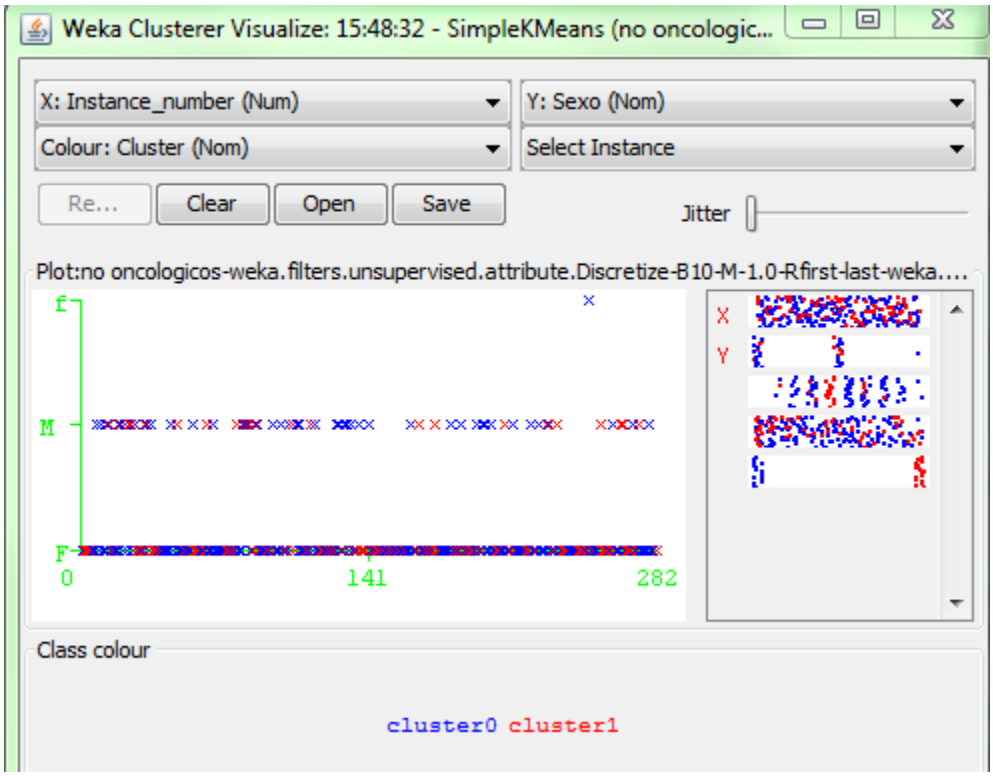
Algoritmo EM: El algoritmo EM asigna a cada instancia una distribución de probabilidad de pertenencia a cada clúster. El algoritmo puede decidir cuántos clústeres crear basado en validación cruzada o se le puede especificar a priori cuantos debe generar. Utiliza el modelo Gaussiano finito de mezclas, asumiendo que todos los atributos son variables aleatorias independientes. Este algoritmo es bastante más elaborado que el K- Medias, ya que requiere muchas más operaciones.

Desarrollo

Del archivo de tipos de cáncer proporcionado, se realizara el agrupamiento respectivo, es importante que el alumno siga el proceso para obtener resultados significativos y que reporte lo obtenido.

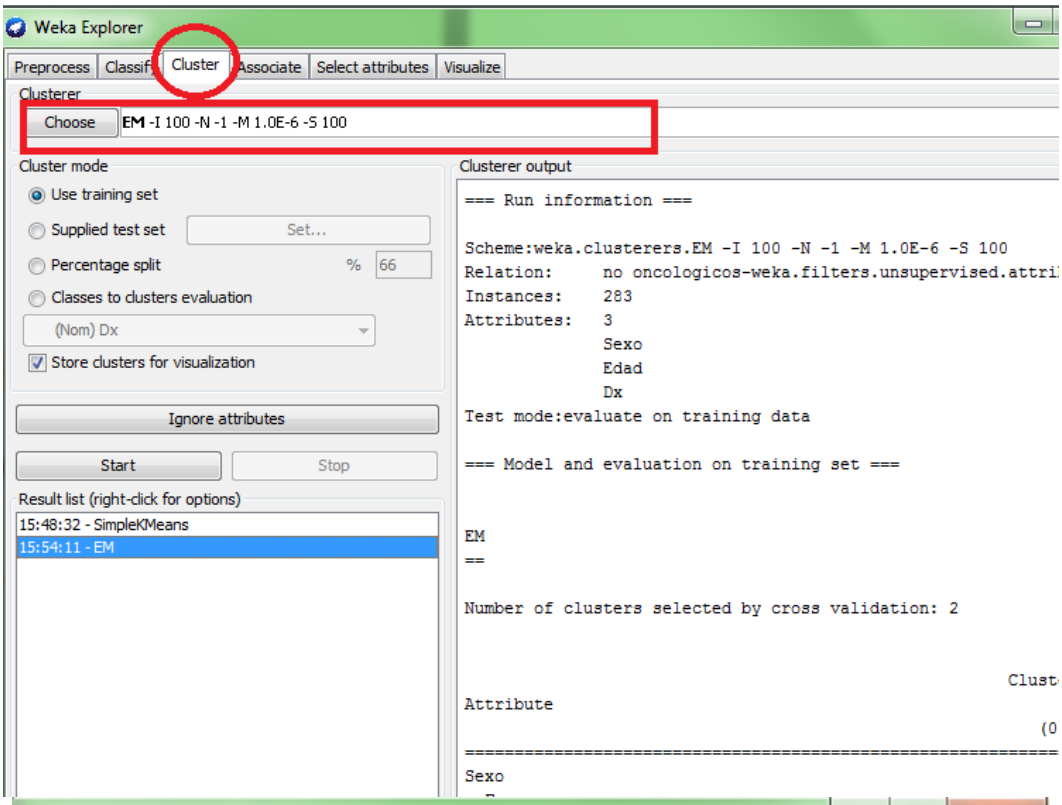
1. Aplicar el **Algoritmo K-Medias** al banco de datos

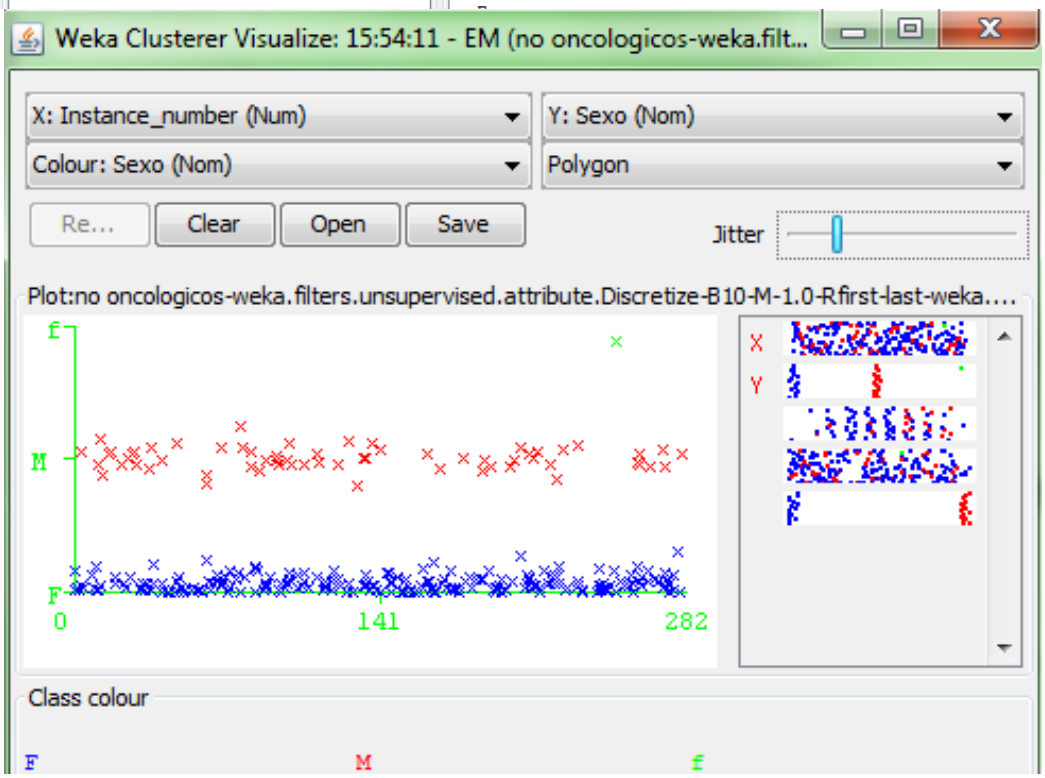




2. Indique los principales resultados obtenidos

3. Aplicando el **Algoritmo EM**





Indique los principales resultados obtenidos

Toda la documentación elaborada deberá subirla a su portafolio de SEDUCA en la fecha indicada.

Conclusiones

Anote de manera breve las principales conclusiones obtenidas al término de esta práctica

Bibliografía

Básico:

Hilera Gonzales, J. R. & Martinez Hernando V. J,(Eds.2005) Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones, ra-ma (Libro),ISBN 84-7897-155-6, Madrid, España

Anderson, J. A. & Rosenfeld, E. (Eds.) (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences, 79, 2554-2558.

Freeman, J.A. & Skapura, D. M (1992). Neural Networks: Algorithms, Applications, and Programming Techniques, Addison-Wesley, Massachusetts.

Complementaria

An introduction to Genetic Algorithms. Autor: Melanie Michell. Editorial: MIT Press

Practical Genetic Algorithms. Randy l Haup, sue Ellen Haup Ed.: Wiley

Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975, 211 p.

Koza, J.R., "Genetic Programming. On the Programming of Computers by Means of Natural Selection", The MIT Press, 1992, 819 p.