

TG3 – MINERÍA DE DATOS

GRUPO T5

Contenido

1. Autores del trabajo, planificación y entrega	2
1.1 Autores	2
1.2 Planificación	3
1.3 Entrega	3
2. Requisitos del prototipo a implementar	4
2.1 Requisitos funcionales	4
2.2 Otros requisitos	5
3. Criterios de comparación en la implementación	6
3.1 Criterio 1: Tiempo	6
3.2 Criterio 2: Funcionalidad	7
3.3 Criterio 3: Diseño	8
4. Proyecto de implementación de un prototipo del sistema utilizando Rapidminer	12
4.1 Documentación de diseño	12
4.2 Documentación de construcción	14
4.3 Documentación de pruebas	15
4.4 Documentación de instalación	18
4.5 Manual de usuario	19
5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B	21
5.1 Documentación de diseño	21
5.2 Documentación de construcción	21
5.3 Documentación de pruebas	22
5.4 Documentación de instalación	24
5.5 Manual de usuario	26
6. Comparación de las dos implementaciones	30
6.1 Evaluación de los criterios en la implementación usando RapidMiner	30
6.2 Evaluación de los criterios en la implementación usando Weka Oracle	34
7. Comparación de la implementación de las tecnologías	38
3.1 Criterio 1: Tiempo	38
3.2 Criterio 2: Funcionalidad	39
3.3 Criterio 3: Diseño	39
8. Conclusiones	42

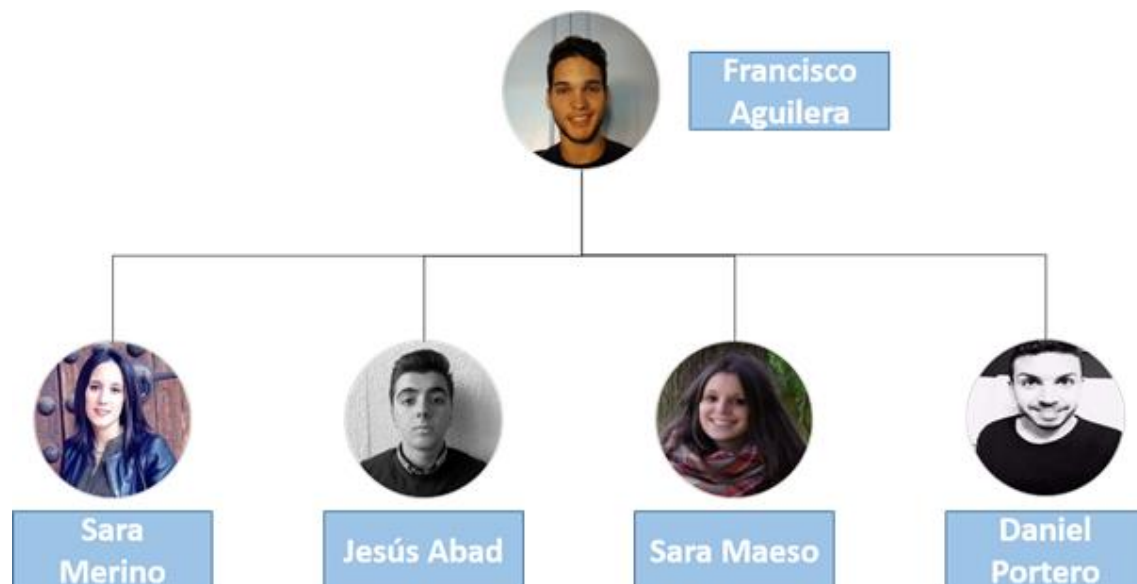
1. Autores del trabajo, planificación y entrega

1. Autores del trabajo, planificación y entrega

1.1 Autores

Los autores de este trabajo (TG1) son los siguientes:

- Francisco José Aguilera Matas – Coordinador
- Jesús Abad LaFuente
- Sara Maeso Cárdbaba
- Sara Merino Benito
- Daniel Portero Carrasco



1.2 Planificación

La planificación de este trabajo se ha realizado por el Coordinador del grupo a través de diferentes plataformas. Lo primero que se ha realizado es una planificación y reparto de las actividades a través de la plataforma online **GanttPro**, para poder ver el contenido de las tareas con sus planificaciones horarias y sus asignaciones pinche [aquí](#).

Una vez se ha realizado el GanttPro se ha creado un repositorio en la plataforma **GitHub** donde cada miembro del grupo ha tenido que subir sus partes correspondientes. Para poder ver el repositorio pinche [aquí](#).

Enlace al documento word en línea de Google drive: [Enlace](#).

Además se han utilizado plataformas como Google HangGouts para realizar videoconferencias entre los miembros del grupo y el coordinador para la planificación y para ver el avance del proyecto.

Otras plataformas utilizadas: Google Drive (Google docx, Google pptx) para la creación de los documentos y las presentaciones, Whatsapp comunicación rápida del equipo.

1.3 Entrega

Enlace al repositorio GitHub: [Enlace](#).

2. Requisitos del prototipo a implementar

2.1 Requisitos funcionales

En este apartado se recogerán todos los requisitos funcionales que deberán poseer los programas de minería de datos para las dos implementaciones, los requisitos se mostrarán en una tabla en la que se mostrará el número de referencia, el nombre, una descripción y la prioridad que posee este requisito que oscila entre 1-5.

REQ.	NOMBRE	DESCRIPCIÓN	PRIORIDAD
RF01	PLATAFORMAS	El programa debe poder correrse en las siguientes plataformas: Microsoft Windows, Mac OSX o Linux.	4
RF02	LENGUAJE	Debe admitir el lenguaje JAVA.	4
RF03	FORMATO DE LOS DATOS	Es necesario que el programa pueda leer datos en formato CSV separado por comas, ya que las BBDD se guardarán siempre en este formato.	5
RF04	UTILIDAD DEL PROGRAMA PARA LA BBDD	El programa mostrará la BBDD, y además de ello nos deberá permitir modificarla, ya sean valores concretos o las columnas (ya sea el nombre o su valor: int, char, date...)	4
RF05	ENGANCHAR DATOS Y ALGORITMO	Se pide que el programa pueda enganchar directamente a la colección de datos cargada un algoritmo de aprendizaje que nos genere un modelo clasificador automático.	4
RF06	ÁRBOLES DE DECISIÓN	Debe poder realizarse árboles de decisiones para poder obtener un mejor entendimiento de los datos.	5
RF07	RESULTADOS	Es necesario que los resultados de la evaluación que nos permite ver cómo de buenas son las clasificaciones de nuestro clasificador automático se muestren.	5
RF08	EXTENSIÓN 1	Se pide que el programa pueda instalar la extensión o plugin de Text Processing.	4

RF09	EXTENSIÓN 2	Se pide que el programa pueda instalar la extensión o plugin de Web Mining	4
------	-------------	--	---

2.2 Otros requisitos

En este apartado se recogerán todos los requisitos no funcionales que deberán poseer los programas de minería de datos para las dos implementaciones, los requisitos se mostrarán en una tabla en la que se mostrará el número de referencia, el nombre, una descripción y la prioridad que posee este requisito que oscila entre 1-5.

REQ.	NOMBRE	DESCRIPCIÓN	PRIORIDAD
RNF01	INTERFAZ DE USUARIO	La interfaz de usuario debe poseer botones intuitivos para que la persona que lo use pueda realizar su trabajo de manera más sencilla.	2
RNF02	EXTENSIÓN 3	Weka Extension.	3
RNF03	VISUALIZACIÓN DE DATOS	El programa mostrará el resultado de los datos en gráficos.	3
RNF04	HERRAMIENTA	La herramienta debe ser preferentemente Open Source.	2

3. Criterios de comparación en la implementación

3.1 Criterio 1: Tiempo

3.1.1 Criterio de tiempo de planificación

Nombre del criterio: Tiempo de planificación

Descripción: Horas invertidas en la planificación inicial antes de implementar el prototipo.

Tipo de valor: Numérico (horas)

3.1.2 Criterio de tiempo de preparación

Nombre del criterio: Tiempo de preparación

Descripción: Horas invertidas en la visualización de tutoriales, documentos o archivos para lograr a comprender correctamente el programa antes de su uso.

Tipo de valor: Numérico (horas)

3.1.3 Criterio de tiempo para la organización de los recursos

Nombre del criterio: Organización de los recursos

Descripción: Tiempo que se ha tardado en la descripción de los recursos iniciales para su implementación.

Tipo de valor: Numérico (horas)

3.1.4 Criterio de tiempo de instalación de la tecnología a usar

Nombre del criterio: Instalación de software

Descripción: Tiempo que se ha tardado en la instalación del software.

Tipo de valor: Numérico (horas)

3.1.5 Criterio de tiempo de ajuste del sistema

Nombre del criterio: Ajuste del sistema

Descripción: Tiempo que se ha tardado en ajustar el sistema al nuevo programa

Tipo de valor: Numérico (horas)

3.1.6 Criterio de tiempo de pruebas

Nombre del criterio: Pruebas de la implementación

Descripción: Tiempo que se ha tardado en realizar todas las pruebas hasta el resultado final de la implementación

Tipo de valor: Numérico (horas)

3.1.7 Criterio de velocidad de funcionamiento del sistema

Nombre del criterio: Velocidad de funcionamiento del sistema

Descripción: Tiempo que se tarda en ejecutar el resultado final hasta que se muestra por pantalla

Tipo de valor: Numérico (horas)

3.2 Criterio 2: Funcionalidad

3.2.1 Criterio de facilidad de uso

Nombre del criterio: Facilidad de uso

Descripción: esfuerzo realizado para lograr el resultado final

Tipo de valor: Numérico (del 1 al 10)

3.2.2 Criterio de flexibilidad

Nombre del criterio: Flexibilidad

Descripción: capacidad de adaptación a diversos problemas para acomodar el trabajo

Tipo de valor: Numérico (del 1 al 10)

3.2.3 Criterio de claridad

Nombre del criterio: Claridad

Descripción: facilidad de entender el funcionamiento del programa

Tipo de valor: Numérico (del 1 al 10)

3.2.4 Criterio de documentación

Nombre del criterio: Documentación de soporte

Descripción: disponibilidad de manuales, guías o cualquier tipo de documento para facilitar el uso del programa

Tipo de valor: Numérico (del 1 al 10)

3.2.5 Criterio de recuperabilidad

Nombre del criterio: Recuperación de datos

Descripción: posee utilidades de backup o restore

Tipo de valor: booleano (sí / no)

3.2.6 Criterio de seguridad

Nombre del criterio: Seguridad

Descripción: capacidad de ingresar al programa mediante usuario y contraseña

Tipo de valor: booleano (sí / no)

3.3 Criterio 3: Diseño

3.3.1 Criterio de control de usuario

Nombre del criterio: Criterio de control de usuario

Descripción: Un buen diseño debe estar direccionado a soportar el hecho de que el usuario es quien tiene el control en la GUI. El usuario tiene la libertad para moverse de ventana a ventana y hacer cualquier cosa que desee.

Tipo de valor: booleano (sí / no)

3.3.2 Criterio de control de Sensibilidad

Nombre del criterio: Sensibilidad

Descripción: El sistema debe proporcionar retroalimentación tangible e inmediata para cada acción. Se deben usar cuadros de diálogo para indicar errores de usuario, a través de mensajes claros y entendibles. Nunca mensajes generados por el sistema operativo

Tipo de valor: booleano (sí / no)

3.3.3 Criterio de control de Personalización

Nombre del criterio: Personalización

Descripción: Se debe permitir personalizar las diferentes ventanas del sistema, a los diversos tipos de usuarios que las acceden, teniendo cuidado de modificar algunos aspectos como colores, ocultamiento de columnas, etc.

Tipo de valor: booleano (sí / no)

3.3.4 Criterio de control de Dirección

Nombre del criterio: Dirección

Descripción: Se debe tener presente que la memorización de comandos no aplica bajo GUI. Especialmente el hecho de ubicar un objeto en el sistema, debe ser tan intuitivo como señalarlo con el mouse y realizar la operación deseada con el objeto. Se pueden usar para tal propósito iconos y barras de herramientas que aclaren la ubicación de los diferentes objetos existentes.

Tipo de valor: booleano (sí / no)

3.3.5 Criterio de control de Consistencia

Nombre del criterio: Consistencia

Descripción: El sistema deberá ser consistente con el mundo en que los usuarios viven y trabajan diariamente. Para ello se debe usar el vocabulario que manejan los usuarios y tratar de estandarizarlo a lo largo de todo el sistema, para que la GUI sea entendible por ellos.

Una clave aquí de estándares, es tratar de mantener los definidos en aplicaciones de uso general como Word y Excel, que siempre tratan de mostrar la misma interface para el usuario.

Tipo de valor: booleano (sí / no)

3.3.6 Criterio de control de Claridad

Nombre del criterio: Claridad

Descripción: La información presentada en la interface debe ser inmediatamente comprensible y el uso de la aplicación debe ser visualmente evidente. Se deben usar tablas de control a través de listas desplegables para dar mayor información a los usuarios, cuando se necesitan digitar datos como por ejemplo, sexo, estado civil, departamento, etc.

Tipo de valor: booleano (sí / no)

3.3.7 Criterio de control de Estética

Nombre del criterio: Estética

Descripción: La composición y disposición de una ventana debe ser visualmente agradable. Deberá atraer la vista hacia la información que es más importante. El ojo humano ve primero la parte izquierda superior del centro de la pantalla y hace un barrido en el sentido de las agujas del reloj.

Se debe tener especial cuidado con los colores a usar, el tipo de letra, el tamaño de la misma. No se deben presentar ventanas muy atiborradas de objetos ; es mejor dividir las en otras ventanas, para evitar confusiones.

Tipo de valor: booleano (sí / no)

3.3.8 Criterio de control de Indulgencia

Nombre del criterio: Indulgencia

Descripción: Un buen diseño de interface debe motivar la exploración. El usuario debe sentirse libre para husmear por la aplicación y dar vistazos rápidos en las diversas ventanas y característica. Se debe dar también una forma de salida agradable cuando se decide abandonar ya sea una transacción o la aplicación misma.

Tipo de valor: booleano (sí / no)

3.4 Criterio 4: Calidad

Conjunto de atributos que se relacionan con la capacidad que tiene el software de ser transferido, desde un ambiente a otro.

3.4.1 Criterio de Facilidad de Auditoría

Nombre del criterio: Facilidad de Auditoría

Descripción: La facilidad con que se puede comprobar la conformidad con los estándares.

Tipo de valor: Numérico (del 1 al 10)

3.4.2 Criterio de Seguridad

Nombre del criterio: Seguridad

Descripción: La disponibilidad de mecanismos que controlen o protejan los programas o datos

Tipo de valor: Booleano (Sí / No)

3.4.3 Criterio de Facilidad de Operación

Nombre del criterio: *Facilidad de Operación*

Descripción: *La facilidad de operación de un programa*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.4 Criterio de Completitud

Nombre del criterio: *Completitud*

Descripción: *El grado en que se ha conseguido la total implementación de las funciones requeridas*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.5 Criterio de Concisión

Nombre del criterio: *Concisión*

Descripción: *Lo compacto que es el programa en términos de líneas de código*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.6 Criterio de Consistencia

Nombre del criterio: *Consistencia*

Descripción: *El uso de un diseño uniforme de técnicas de documentación a los largo del proyecto de desarrollo de software*

Tipo de valor: *Booleano (Sí / No)*

3.4.7 Criterio de Auto-Documentación

Nombre del criterio: *Auto-Documentación*

Descripción: *El grado en que el código fuente proporciona documentación significativa*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.8 Criterio de Tolerancia de Errores

Nombre del criterio: *Tolerancia de Errores*

Descripción: *El daño que se produce cuando el programa encuentra un error*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.9 Criterio de Eficiencia en la Ejecución

Nombre del criterio: *Eficiencia en la Ejecución*

Descripción: *El rendimiento en tiempo de ejecución de un programa*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.10 Criterio de Facilidad de expansión

Nombre del criterio: *Facilidad de expansión*

Descripción: *El grado en que se puede ampliar el diseño arquitectónico de datos o procedural*

Tipo de valor: *Numérico (del 1 al 10)*

3.4.11 Criterio de Instrumentación

Nombre del criterio: Instrumentación

Descripción: El grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen

Tipo de valor: Numérico (del 1 al 10)

3.4.12 Criterio de Modularidad

Nombre del criterio: Modularidad

Descripción: La independencia funcional de los componentes del programa

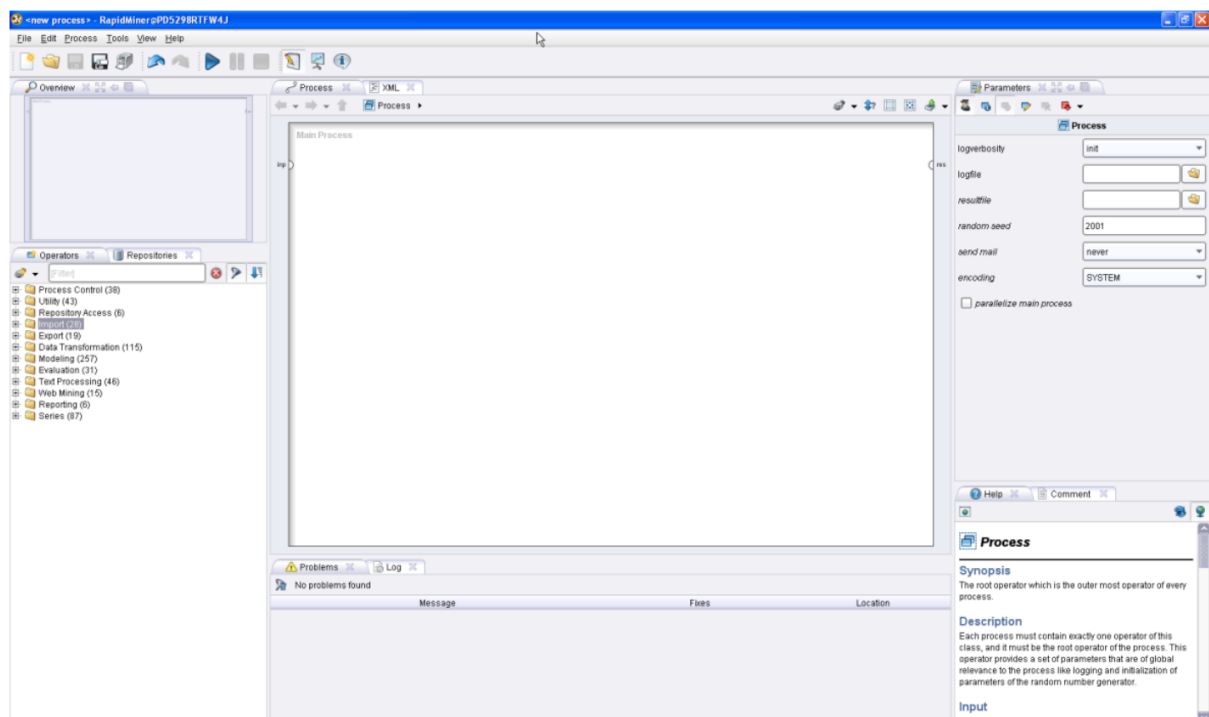
Tipo de valor: Numérico (del 1 al 10)

4. Proyecto de implementación de un prototipo del sistema utilizando Rapidminer

4.1 Documentación de diseño

El Objetivo de este ejercicio es evaluar distintos algoritmos de Aprendizaje Supervisado para tareas de clasificación.

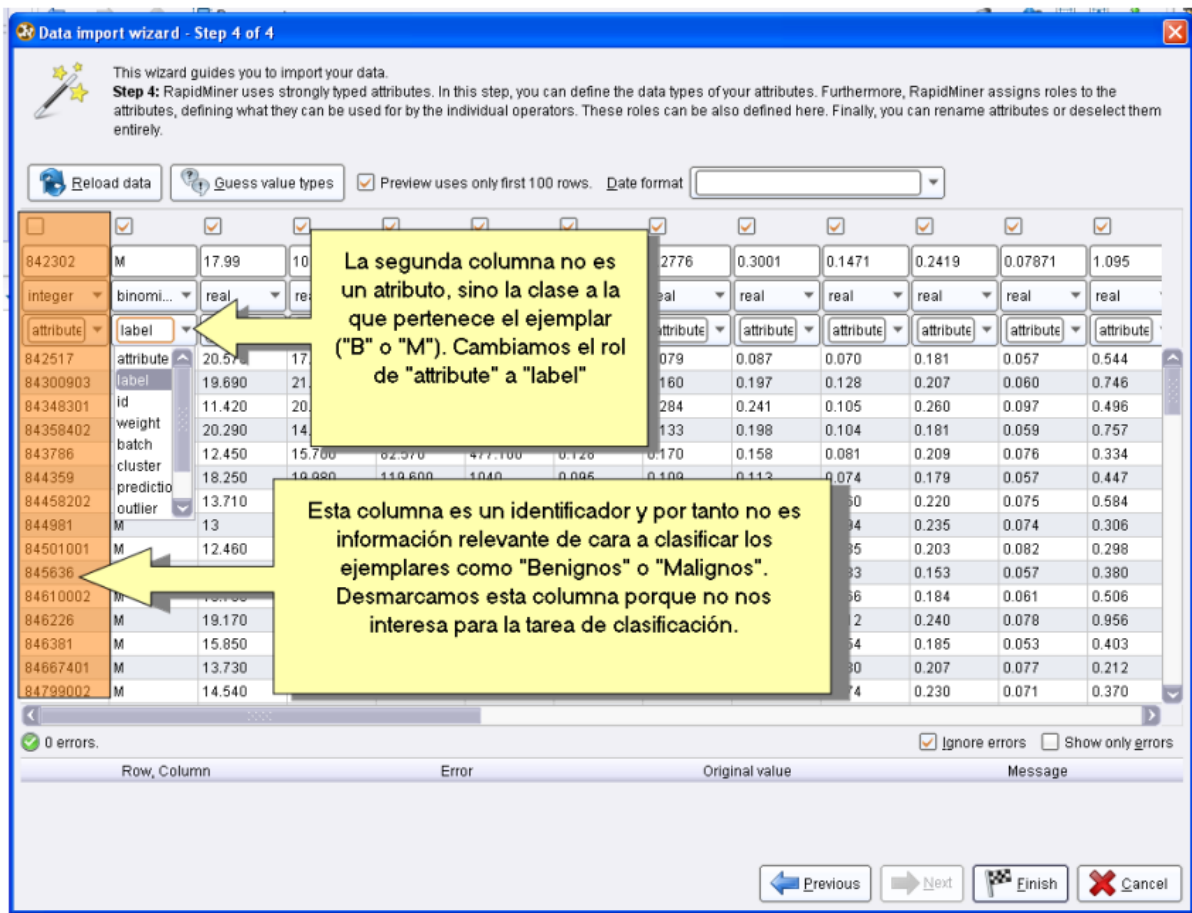
Una vez realizado los pasos de instalación de RapidMiner mostrados en el punto 4.4, la interfaz inicial y donde se va a trabajar aparece a continuación.



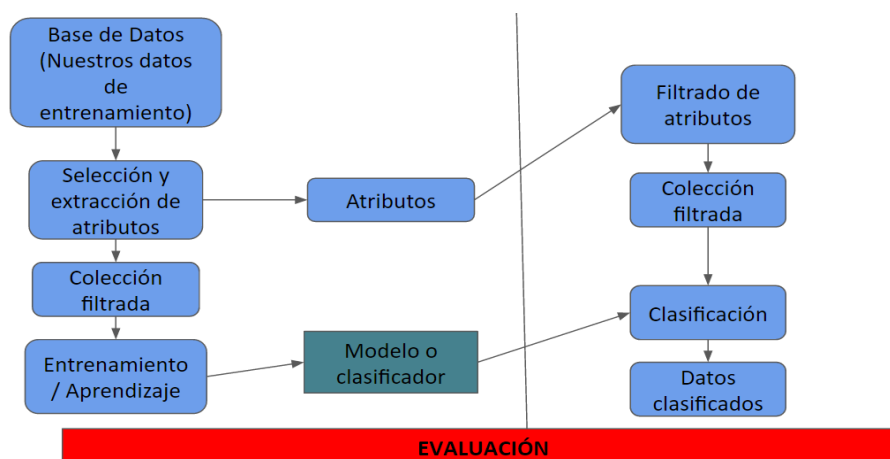
En dicha interfaz se importará el archivo de datos '.data', el cual será leído por nuestro acceso de datos "Read csv". Hay que cambiar el filtro de extensiones a "all files" ya que por defecto espera únicamente archivos ".csv" o ".txt".

Una vez leídos los datos tenemos que indicar cuál es el carácter separador, en este caso delimitado por comas.

Por último tenemos que indicar qué columna es la que contiene la clase de los ejemplares (B o M) en nuestro caso. También definimos de qué tipo son los atributos (numéricos, nominales, binarios, etc).



Una vez cargados los datos correctamente, faltaría realizar el **"Punto 4.4 Resultados"** para mostrar por pantalla los resultados de nuestro archivo de datos.



4.2 Documentación de construcción

Construiremos un sistema para detectar cáncer de mama utilizando una colección de datos formada por casos tratados en los hospitales de la universidad de Wisconsin.

El sistema que vamos a desarrollar será capaz de indicarnos si a tenor de los resultados analíticos de un nuevo caso no diagnosticado, estamos ante un caso “benigno” o “maligno”.

Las siguientes líneas muestran un ejemplo en Java de cómo cargar y usar un clasificador automático:

```
...
Classifier classifier = new J48();
try {
    Trainer trainer = new Trainer();
    classifier = trainer.loadClassifier("data/Model.dat");

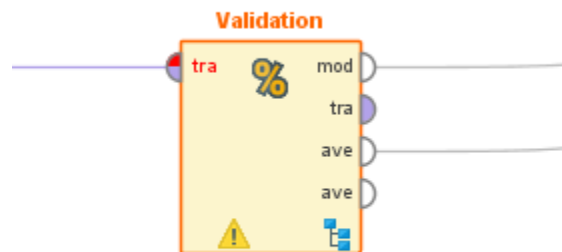
    Instance instance;
    // Aquí iría la parte en la que definimos la nueva instancia a clasificar
    // por ejemplo, si es un documento, lo cargamos y lo tokenizamos

    // La siguiente línea clasifica la nueva instancia
    double clase = classifier.classifyInstance(instance);
    if (clase == 1) {
        System.out.println("La Clase es de tipo 1");
        out.println("Clase 1");
    } else {
        System.out.println("Clase complementaria");
        out.println(codigo.toString());
    }
}
catch (Exception e) { e.printStackTrace(); }
```

4.3 Documentación de pruebas

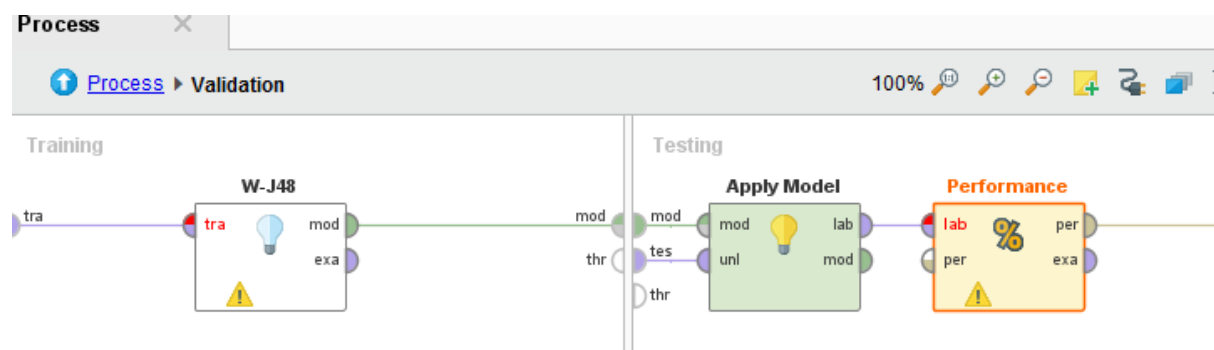
Errores

Algunos de los errores más comunes que hemos tenido a la hora de realizar la implementación han sido a la hora de cargar los datos de la base de datos en formato CSV, ya que no fijamos bien la limitación, está tenía que ser por comas para que se viera correctamente y lo leyera.



Nos salía una advertencia para avisarnos dentro de X-Validation, ya que es el algoritmo que va conectado a nuestra lectura del fichero.

Otros errores, fueron los siguientes:

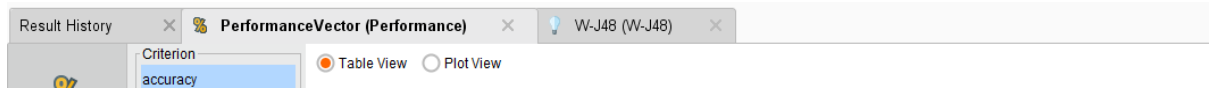


Si hacemos doble click en la caja del proceso de validación cruzada, nuestra interfaz como bien vemos se divide en dos partes. La de nuestra izquierda se corresponde en off-line y la derecha en online. En la derecha nos sale un error en Performance, esto de debía a que no cogíamos bien la caja correcta, sino un clasificador que no hacía bien su función además de los numerosos errores que nos daba porque no leía correctamente los datos de evaluación como bien explicamos antes.

Resultados

Una vez que ya hemos terminado de definir el proceso de generación y evaluación del clasificador automático creado, podemos ejecutarlo pinchando en un triángulo azul que aparece en la parte superior del programa.

Con ello, se abre una vista de resultados donde se nos muestra los elementos que hayamos conectado a la salida.



Aquí podemos ver como en una pestaña tenemos los resultados de la evaluación y el modelo, un árbol de decisión. Estos resultados nos permite ver cómo de buenas son las clasificaciones de nuestro clasificador automático.

accuracy: 94.55% +/- 1.45% (mikro: 94.54%)

	true M	true B	class precision
pred. M	191	11	94.55%
pred. B	20	346	94.54%
class recall	90.52%	96.92%	

La siguiente captura nos muestra una vista de resultados de evaluación, una matriz de confusión, con los falsos positivos y los falsos negativos.

En este ejemplo el algoritmo W-J48 que hemos utilizado clasifica correctamente el 90.52% de los cánceres malignos y el 96.92% de los cánceres benignos.

También nos muestra el árbol de decisión generado visto de dos maneras:

Descripción:

W-J48

J48 pruned tree

```
-----
2019 <= 880.8
| 0.2654 <= 0.1357
| | 153.4 <= 36.46: B (319.0/3.0)
| | 153.4 > 36.46
| | | 17.99 <= 14.97
| | | | 0.9053 <= 1.978: B (11.0)
| | | | 0.9053 > 1.978
| | | | 0.9053 <= 2.239: M (2.0)
| | | | 0.9053 > 2.239: B (3.0)
| | | 17.99 > 14.97: M (2.0)
| 0.2654 > 0.1357
| | 17.33 <= 27.37
```

```

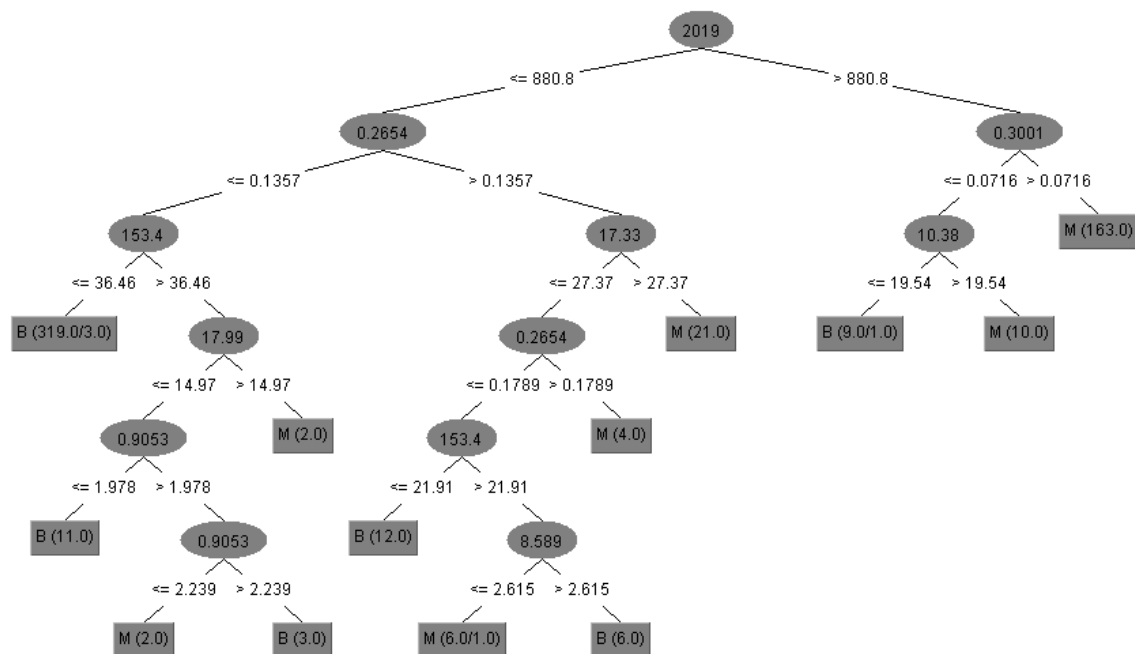
| | | 0.2654 <= 0.1789
| | | | 153.4 <= 21.91: B (12.0)
| | | | 153.4 > 21.91
| | | | | 8.589 <= 2.615: M (6.0/1.0)
| | | | | 8.589 > 2.615: B (6.0)
| | | 0.2654 > 0.1789: M (4.0)
| | 17.33 > 27.37: M (21.0)
2019 > 880.8
| 0.3001 <= 0.0716
| | 10.38 <= 19.54: B (9.0/1.0)
| | 10.38 > 19.54: M (10.0)
| 0.3001 > 0.0716: M (163.0)

```

Number of Leaves : 13

Size of the tree : 25

Weka Result



Si queremos evaluar otros algoritmos de aprendizaje lo que tendríamos que hacer es cambiar la caja de W-J48 por la de otros algoritmos incluidos en la misma carpeta como por ejemplo (Bayes, PRISM...) y comprobar los resultados que obtenemos. Con esta manera lo que conseguimos es comparar los distintos resultados y algoritmos.

4.4 Documentación de instalación

Para realizar este ejercicio lo primero que vamos a realizar es actualizar la versión de RapidMiner que tenemos en el PC, en este caso instalamos la versión RapidMiner 7.1 e instalamos una serie de extensiones que vamos a utilizar.

Para la instalación de RapidMiner acudimos al siguiente enlace que nos deja directamente en la instalación de la versión 7.1:


<https://my.rapidminer.com/nexus/account/index.html#downloads>

Downloads


Click on a RapidMiner product of your choice to download it.

RapidMiner Studio 7.1

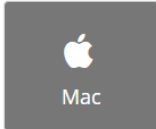
Click on your operating system to start the download:



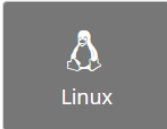
Windows
32bit



Windows
64bit



Mac
Requires: Mac OS 10.8+

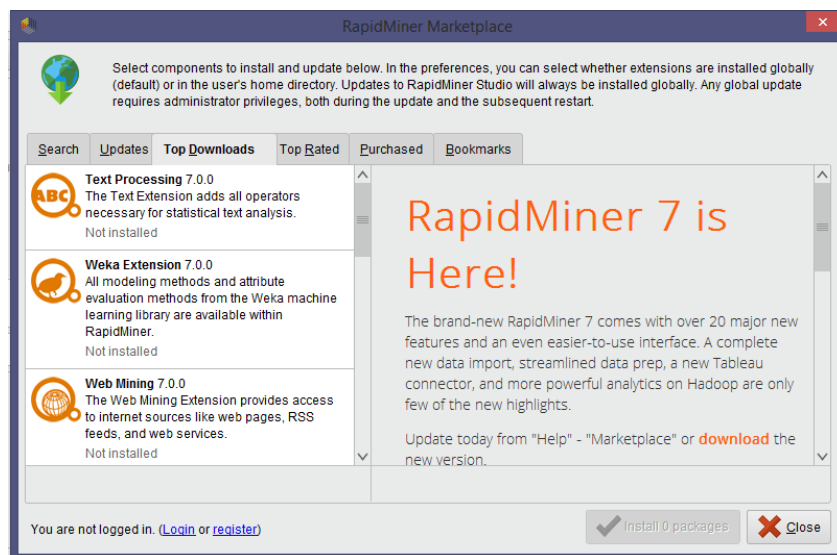


Linux
Requires: Java 8 or Java 7

- [Installation Guide](#)
- [Getting Started Tutorials](#)
- [Support](#)
- [Download Source](#)

Una vez instalado, lo más recomendable es registrarse en la comunidad RapidMiner. Esto te aparecerá nada más ejecutar el programa.

Para instalar las extensiones, así como actualizar el propio RapidMiner, nos vamos a Extensions -> MarketPlace y nos aparecerá el siguiente recuadro.7



Para el siguiente ejercicio de Aprendizaje automático que vamos a realizar necesitaremos las siguientes extensiones, que además son las primeras que aparecen en la pantalla de descargas, en *Top Downloads*, si no podemos buscarlas en la pestaña *Search*:

- **Weka Extension**
- **Text Processing**
- **Web Mining**

4.5 Manual de usuario

1. Abrir RapidMiner

Si es la primera vez que se ejecuta el programa, hay que seguir los siguientes pasos, de lo contrario omitir hasta el 7.

2. RapidMiner mostrará una alerta indicando que no se tiene un repositorio creado, se debe pulsar OK.

3. En la pantalla siguiente, seleccionar New Local repository y hacer click en Siguiente.

4. En el campo de Alias seleccionar un nombre para el repositorio, por ejemplo Local Repository.

5. En el campo Root Directory, seleccionar una carpeta del disco en la que se quiera ubicar el repositorio.

6. Hacer click en Terminado.

En la ventana de bienvenida debería aparecer, mostrando iconos para las opciones New, Open Recent, Open, Open Template y Online Tutorial.

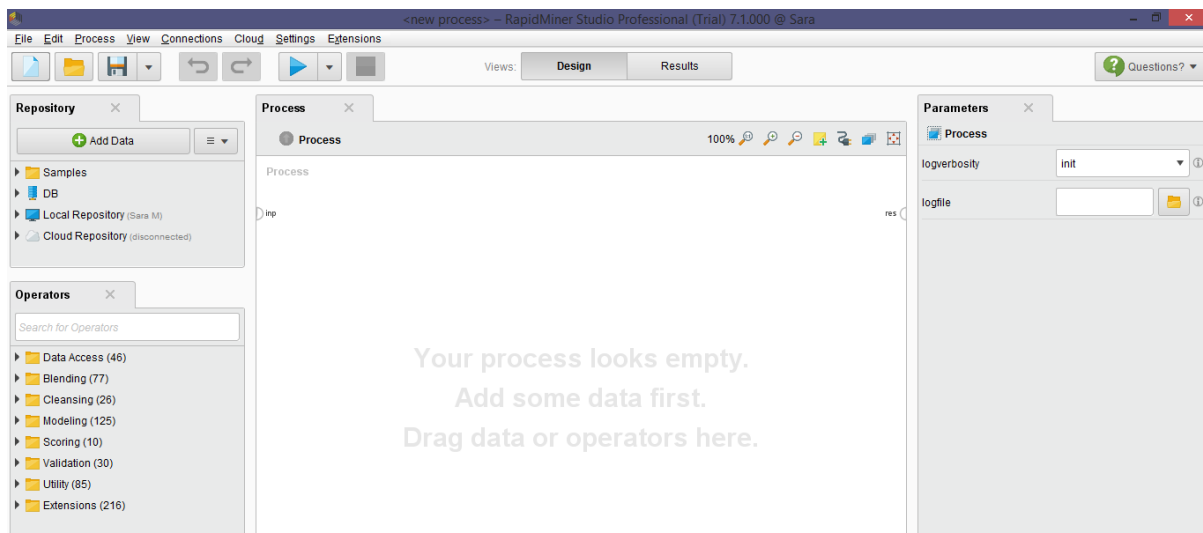
7. Seleccionar New

La ventana de fondo cambiará y sobre ella aparecerá el explorador de repositories (Repository Browser), en esta ventana:

8. Seleccionar el repositorio que se quiere usar (ejemplo: LocalRepository)

9. En el campo Name, introducir un nombre para este proyecto (Ejercicio1).

Si los pasos se han seguido correctamente, se estará observando la perspectiva de diseño de RapidMiner, que se compone básicamente de 3 paneles verticales.



El de más a la izquierda sirve para tener acceso a los operadores y repositorios. El central (y el más grande por defecto), será el área en el que se diseñará el proceso y en el que se ubicaran los operadores que construyan el modelo. Por último, en el panel de la derecha se sitúa la pestaña de parámetros (Parameters), que permite configurar los operadores, y la pestaña de ayuda (Help).

Para comenzar a usarlo, hay que importar los datos:

10. En la pestaña, Operadores, explorar las carpetas Import -> Data
11. Hacer click en el operador "Read CSV" en nuestro caso y arrastrarlo hacia el área de trabajo (Main Process)
12. Incluir los parámetros que se necesiten en la implementación, en nuestro caso un árbol de decisión.
13. En la pestaña Operators explorar las carpetas Modeling->Model Application, seleccionar el operador Apply Model y arrastrarlo al área de testing.
14. En la pestaña Operators explorar las carpetas Evaluation->Performance Measurement, seleccionar el operador Performance y arrastrarlo a continuación del operador Apply Model.
15. Conectar la entrada mod de Apply Model, con mod de Testing y la unl con tes.
16. Conectar la salida lab de Apply Model con la entrada lab de Performance.
17. Conectar la salida per de Performance, con la salida ave de Testing.



Una vez finalizado todo el proceso, pulsar el botón de play para ver los resultados.

5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

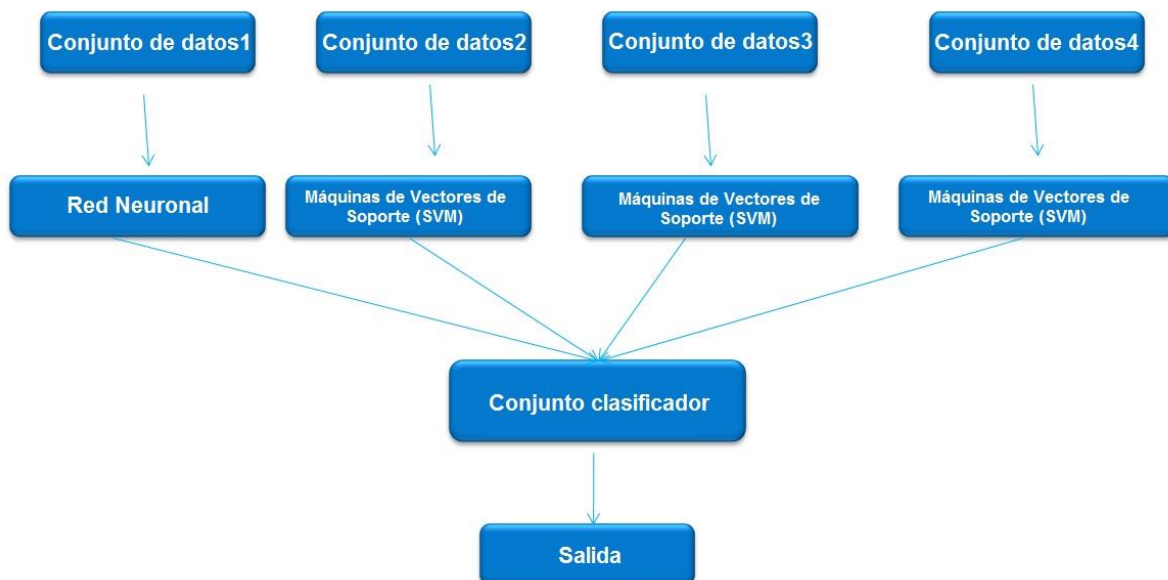
5.1 Documentación de diseño

El programa de implementación cumple con todos los requisitos propuestos anteriormente. El programa Weka Oracle se puede instalar en diferentes sistemas operativos, ya que es multiplataforma, los pasos explicados para su instalación están en el apartado 5.4 Documentación de instalación.

Se utiliza el lenguaje Java.

Para cargar los datos, necesitamos que estos sean de tipo ARFF, DATA, CSV o JSON. Cuando cargamos estos datos, nos muestra las variables y podemos seleccionar tantas variables como tenga el archivo para luego comparar datos y poder analizarlos.

Una vez generado el árbol de decisión y no haya errores en la implementación, este nos mostrará una serie de resultados. Donde se nos permitirá analizar y ver cómo de buenos son los resultados.



5.2 Documentación de construcción

En nuestro caso, implementaremos un ejercicio sobre si se puede jugar o no al tenis, dependiendo de varios factores, como el tiempo, la temperatura, la humedad y si llueve o no.

A continuación, una parte del código del archivo .arff:

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

5.3 Documentación de pruebas

Errores



Error al abrir un fichero.

Resultado

Una vez cargado el archivo.arff y tenemos las variables de “outlook”, “temperature”, “humidity”, “windy” y “play” cargadas, pasamos a clasificar los datos.

Lo que hacemos es seleccionar el algoritmo, en este caso de árbol, el J48, y seleccionamos la variable “play”. El resultado es el siguiente:

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

J48 pruned tree
-----

outlook = sunny: no (5.0/2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves    :    4

Size of the tree    :    6

Time taken to build model: 0.02 seconds
```

Lo que nos dice este resultado es que si el clima es soleado, no juegan 2 de 5 personas; si está nublado, juegan 4 personas; y si llueve, dependiendo si hace o no aire, no juegan 2 personas y juegan 3 respectivamente. También nos muestra el número de hojas (4) y el tamaño del árbol (6).

Otros de los datos que nos muestran es la matriz de confusión:

=== Confusion Matrix ===

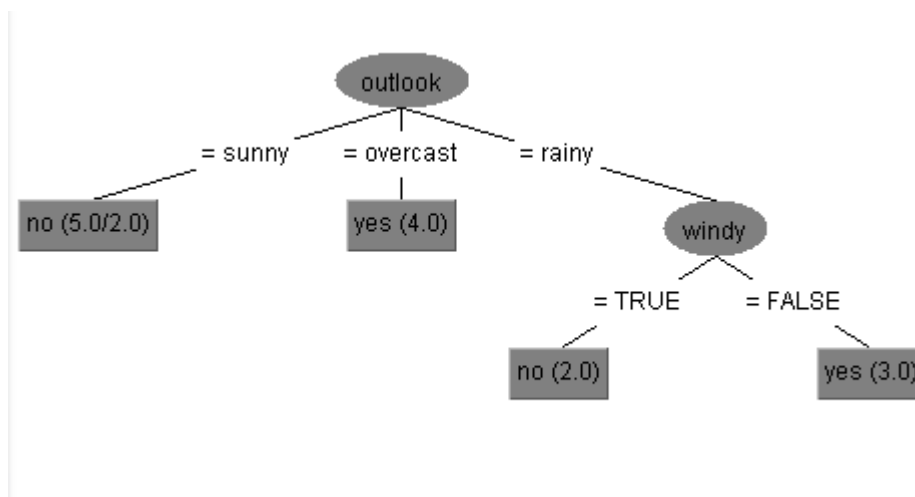
```
a b  <-- classified as
7 2 | a = yes
0 5 | b = no
```

En este caso vemos que la variable “a” son las personas que sí juegan, mientras la “b” son las personas que no juegan.

Nos dice que de 7 registros que se analizan de personas que juegan, hay 0 errores. Y de 5 registros que se analizan de personas que no juegan, hay 2 errores.

En la matriz de confusión debemos tener en cuenta que la diagonal debe de ser mayor, este caso, 7 es mayor que 0, y 5 es mayor que 2. Por lo tanto, la matriz tiene sentido.

La forma gráfica de visualizar el árbol es la siguiente:



5.4 Documentación de instalación

Para realizar este ejercicio lo primero que vamos a realizar es actualizar la versión de Weka Oracle que tenemos en el PC, en este caso instalamos la versión weka 3.8.

Para la instalación de Weka Oracle acudimos al siguiente enlace que nos deja directamente en la instalación de la versión 7.1

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



Machine Learning Group at the University of

[Project](#) [Software](#) [Book](#) [Publications](#) [People](#) |

Weka 3: Data Mining Software in Java

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced **this**, and the bird sounds like **this**.

Weka is open source software issued under the **GNU General Public License**.

We have put together several free online courses that teach machine learning and data mining using WEKA. Check out the **website for the courses** for video lectures and details on how to enrol.

Yes, it is possible to apply Weka to **big data**!

Getting started	Further information	Developers
<ul style="list-style-type: none">• Requirements• Download• Documentation• FAQ• Getting Help	<ul style="list-style-type: none">• Citing Weka• Datasets• Related Projects• Miscellaneous Code• Other Literature	<ul style="list-style-type: none">• Development• History• Subversion• Contributors

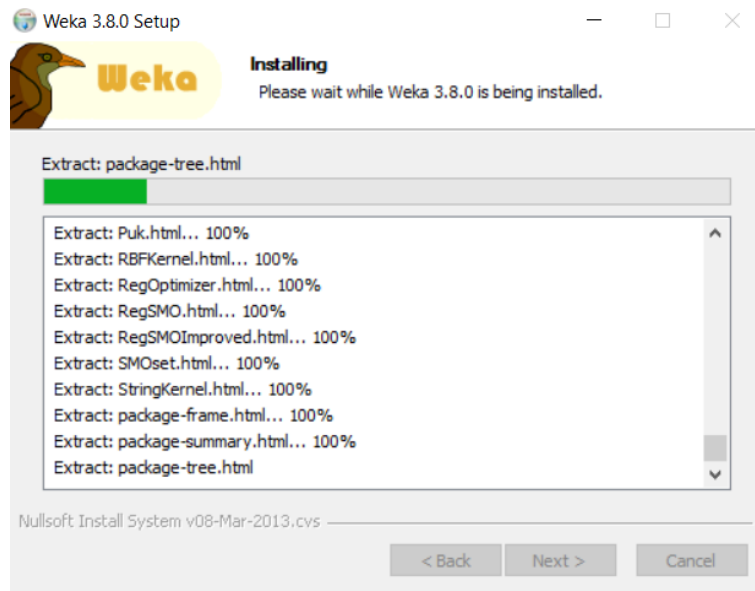
• Stable version

Weka 3.8 is the latest stable version of Weka. This branch of Weka receives bug fixes only, although new features may become available in packages. There are different options for downloading and installing it on your system:

◦ Windows

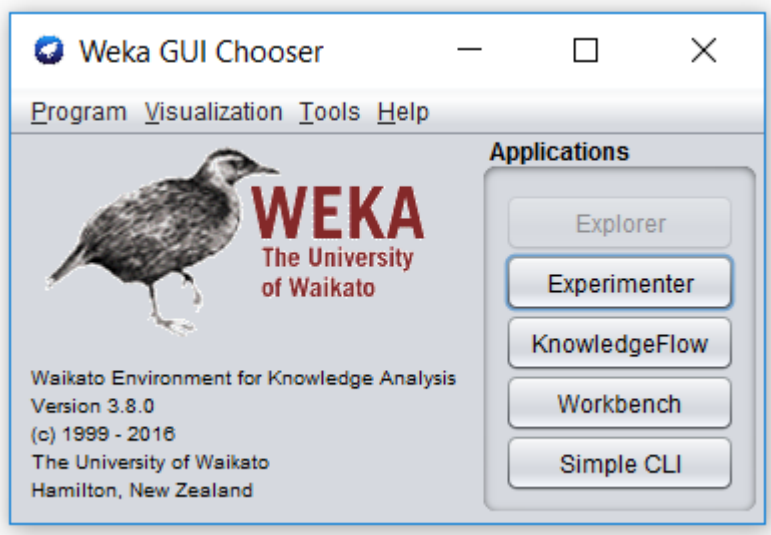
Click [here](#) to download a self-extracting executable for 64-bit Windows that includes Oracle's 64-bit Java VM 1.8 (weka-3-8-0jre-x64.exe; 105.5 MB)

La instalación de weka Oracle es muy sencilla, ya que no requiere la instalación de ninguna extensión. Solamente bajarse el programa y ejecutarlo.



5.5 Manual de usuario

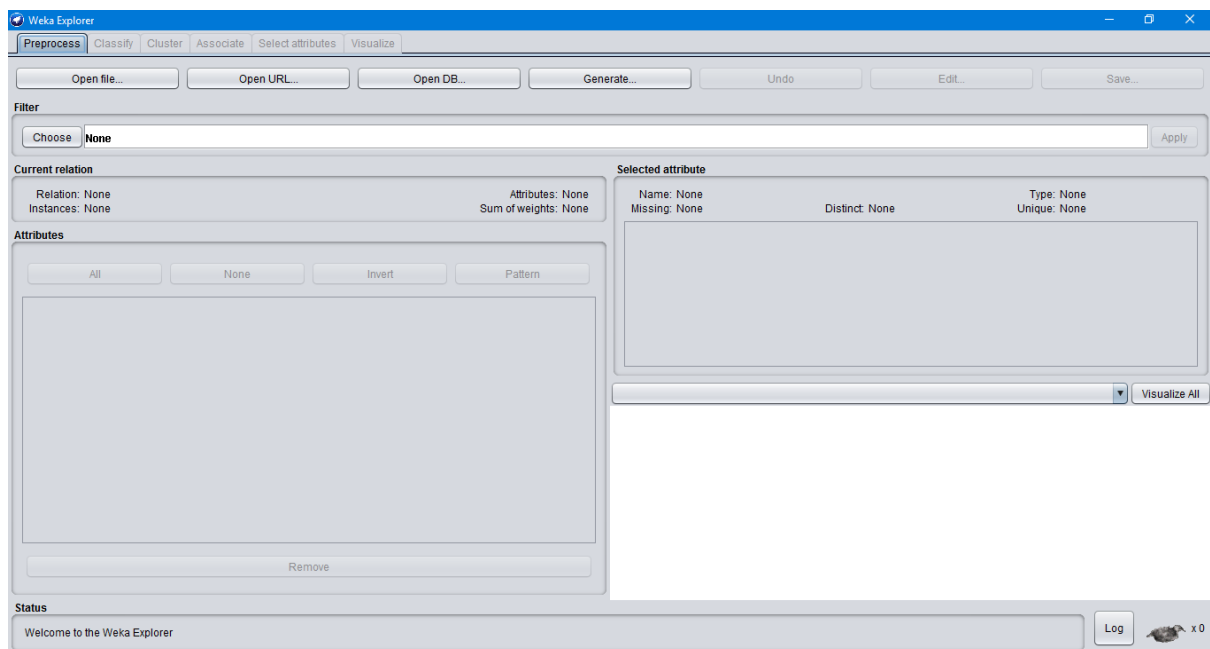
Una vez que Weka Oracle esté en ejecución aparecerá una ventana denominada selector de interfaces, que nos permite seleccionar la interfaz con la que deseemos comenzar a trabajar con Weka Oracle . Las posibles interfaces a seleccionar son Simple Cli, Explorer, Experimenter y Knowledge flow que se explicarán detenidamente y de forma individual en secciones siguientes.



Explorer

El Explorer permite visualizar y aplicar distintos algoritmos de aprendizaje a un conjunto de datos. Cada una de las tareas de minería de datos viene representada por una pestaña en la parte superior. Estas son:

- **Preprocess:** visualización y preprocesado de los datos (aplicación de filtros)
- **Classify:** Aplicación de algoritmos de clasificación y regresión
- **Cluster:** Agrupación
- **Associate:** Asociación
- **Select Attributes:** Selección de atributos
- **Visualize:** Visualización de los datos por parejas de atributos



El primer paso para comenzar a trabajar con el explorador es definir el origen de los datos. Weka Oracle soporta diferentes fuentes que coinciden con los botones que están debajo de las pestañas superiores mostrados en la ventana 4. Las diferentes posibilidades son las siguientes:

Open File

Al pulsar sobre este botón aparecerá una ventana de selección de fichero. Aunque el formato por defecto de Weka Oracle es el arff eso no significa que sea el único que admita, para ello tiene interpretadores de otros formatos. Éstos son:

CSV Archivos separados por comas o tabuladores. La primera línea contiene los atributos.

C4.5 Archivos codificados según el formato C4.5. Unos datos codificados según este formato estarían agrupados de tal manera que en un fichero .names estarían los nombres de los atributos y en un fichero .data estarían los datos en sí.

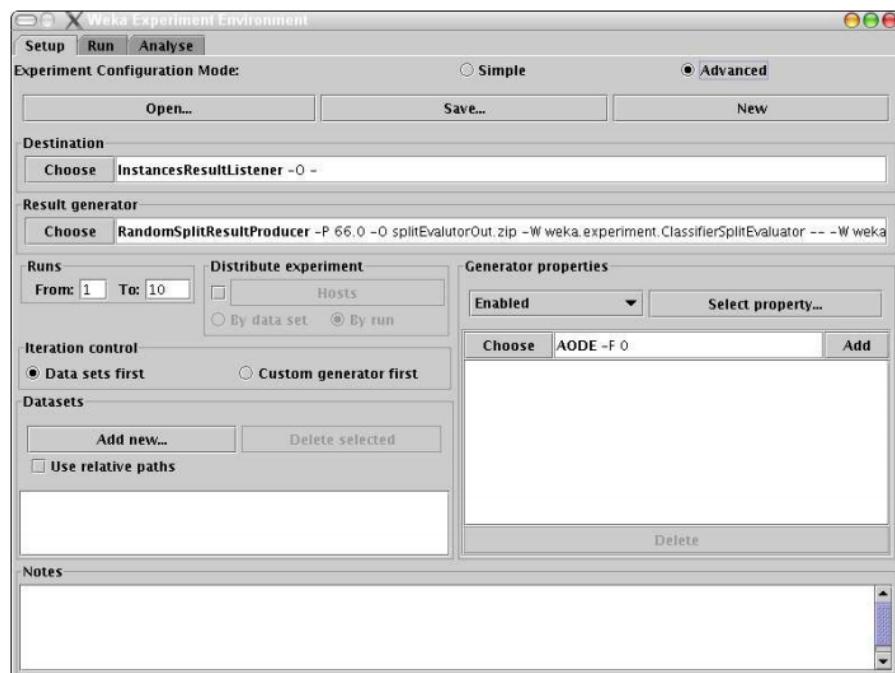
Instancias Serializadas Weka Oracle internamente almacena cada muestra de los datos como una instancia de la clase instance. Esta clase es serializable* por lo que estos objetos pueden ser volcados directamente sobre un fichero y también cargados de uno.

Para cargar un **archivo arff** simplemente debemos buscar la ruta donde se encuentra el fichero y seleccionarlo.

Pulsando en Use converter nos dará la opción de usar un interpretador de ficheros de los tipos ya expuestos. Open Url Con este botón se abrirá una ventana que nos permitirá introducir una dirección en la que definir dónde se encuentra nuestro fichero. El tratamiento de los ficheros (restricciones de formato, etc.) es el mismo que el apartado anterior.

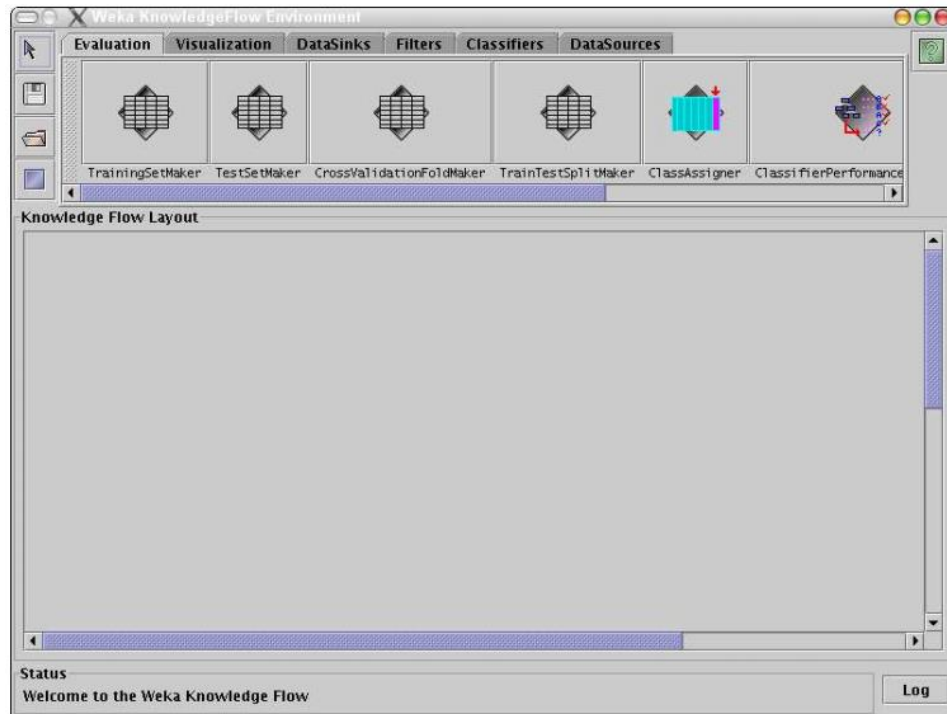
EXPERIMENTER

La principal diferencia es que el funcionamiento de este modo está orientado a realizar tareas específicas más concretas que un experimento normal, y una cierta funcionalidad existente en el modo simple se ha trasladado al modo avanzado, mostrándola más concreta y explícita al usuario.



Knowledge flow

Esta última interface de Weka Oracle es quizá la más cuidada y la que muestra de una forma más explícita el funcionamiento interno del programa. Su funcionamiento es gráfico y se basa en situar en el panel de trabajo, elementos base (situados en la parte superior de la ventana) de manera que creemos un “circuito” que defina nuestro experimento.



6. Comparación de las dos implementaciones

6.1 Evaluación de los criterios en la implementación usando RapidMiner

CRITERIOS DE TIEMPO	EVALUACIÓN
Tiempo de planificación	14 min
Tiempo de preparación	1h 15 min
Tiempo para la organización de los recursos	25 min
Tiempo de instalación de la tecnología a usar	10 min
Tiempo de ajuste del sistema	7 min
Tiempo de pruebas	1h 50 min
Velocidad de funcionamiento del sistema	10 seg en realizar una acción o menos

CRITERIOS DE FUNCIONALIDAD	EVALUACIÓN
Facilidad de uso	8
Flexibilidad	6
Claridad	9
Documentación	10
Recuperabilidad	Sí
Seguridad	Sí

En cuanto a la facilidad de uso, me parece muy intuitivo el programa a la hora de implementar un ejercicio. Para buscar cualquier algoritmo o clase tienes un buscador y no tienes que estar partiendo de carpetas para buscar lo que necesitas.

La flexibilidad, en cuanto a la hora de adaptarse a los diferentes problemas que tuvimos no fue muy bueno.

El programa es bastante claro, tiene bien estructurado su diseño. Se encuentra rápidamente lo que buscas. Te permite una visión clara del ejercicio que estás realizando, así como una buena interfaz en los resultados finales.

La documentación acerca de esta herramienta es muy extensa, tienes muchos manuales y tutoriales para guiarte. Ya bien sea desde la propia página de rapidminer, como de otras personas que comparten estos materiales.

La recuperabilidad del programa es buena, hace backups y te permite recuperar información.

El programa al usarlo y poder acceder a tus actividades creadas te pide un usuario y contraseña, este usuario y contraseña es necesario para poder empezar a utilizar el programa.

CRITERIOS DE DISEÑO	EVALUACIÓN
Control de usuario	Sí
Sensibilidad	Sí
Personalización	Sí
Dirección	Sí
Consistencia	Sí
Claridad	No
Estética	No
Indulgencia	Sí

El control de usuario tiene un buen diseño para soportar el control en la interfaz gráfica de usuario.

La sensibilidad del sistema proporciona cuadros de diálogo para indicar errores de usuario, a través de mensajes claros y entendibles.

La personalización del sistema se permite personalizar las diferentes ventanas del sistema.

La dirección tiene presente que la memorización de comandos no aplica bajo la interfaz gráfica de usuario.

La consistencia del sistema es consistente con el mundo en que los usuarios viven y trabajan diariamente. Para ello el sistema usa el vocabulario que manejan los usuarios y tratar de estandarizar.

La claridad presentada en la interface no es inmediatamente comprensible sin conocimientos previo.

La estética no atrae la vista hacia la información que es más importante.

Sobre la Indulgencia el usuario se siente libre para husmear por la aplicación y dar vistazos rápidos en las diversas ventanas y característica.

CRITERIOS DE CALIDAD	EVALUACIÓN
Facilidad de Auditoría	9
Seguridad	Sí
Facilidad de operar	9
Compleitud	9
Concisión	6
Consistencia	Sí
Auto-Documentación	7
Tolerancia de errores	8
Eficiencia en la ejecución	10
Facilidad de expansión	10
Instrumentación	8
Modularidad	7

Elevada facilidad de Auditoría para poder comprobar la conformidad con los estándares.

Notoria seguridad con diferentes mecanismos que controlen o protejan los programas o datos.

Es una herramienta bastante sencilla para operar con ella.

Hemos conseguido implementar todas las funciones requeridas.

Trabajo bastante compacto referido en términos de líneas de código.

Poca consistencia para el uso de un diseño uniforme de técnicas de documentación a lo largo del proyecto de desarrollo de software.

Se proporciona bastante documentación significativa, lo que nos ha sido bastante útil a la hora de la implementación.

Tiene un alto grado de tolerancia cuando el programa encuentra un error.

Magnífico en tiempo de ejecución de un programa.

Alta facilidad de expansión en el grado en que se puede ampliar el diseño arquitectónico de datos.

Elevado grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen.

Independencia funcional de los componentes del programa normal.

6.2 Evaluación de los criterios en la implementación usando Weka Oracle

CRITERIOS DE TIEMPO	EVALUACIÓN
Tiempo de planificación	20 minutos
Tiempo de preparación	2 horas
Tiempo para la organización de los recursos	20 minutos
Tiempo de instalación de la tecnología a usar	8 minutos
Tiempo de ajuste del sistema	0 minutos
Tiempo de pruebas	2 horas
Velocidad de funcionamiento del sistema	segundos

CRITERIOS DE FUNCIONALIDAD	EVALUACIÓN
Facilidad de uso	7
Flexibilidad	6
Claridad	6
Documentación	8
Recuperabilidad	Sí
Seguridad	No

En cuanto a la facilidad de uso, es bastante intuitivo, pero tiene algunos comandos difíciles de entender y usar.

Es un programa, que si no has visto alguna guía o manual para seguir los pasos, es bastante complejo para analizar datos.

En la recuperabilidad, facilita la recuperación de datos gracias al envío simultáneo de los datos a la base de datos de Oracle mientras vas trabajando.

En Internet podemos encontrar bastantes guías y manuales tanto de instalación de la herramienta como de ejercicios para seguir lo pasos.

Es bastante seguro, ya que puedes controlar quién accede a tus proyectos, dónde y en qué momento. Por lo tanto, tu cuenta quedará bastante protegida.

CRITERIOS DE DISEÑO	EVALUACIÓN
Control de usuario	Sí
Sensibilidad	Sí
Personalización	No
Dirección	Sí
Consistencia	Sí
Claridad	Sí
Estética	Sí
Indulgencia	Sí

En cuanto al control de usuario, tiene libertad de moverse por la herramienta de ventana en ventana.

En la sensibilidad, te muestra errores a la hora de, por ejemplo, cargar archivos que no sean compatibles con el programa.

A la hora de personalizar, la herramienta es bastante estricta y no permite modificar la interfaz.

En la dirección, nos deja ubicar un objeto en el sistema, ya que es intuitivo porque nos deja señalar con el mouse, y además, podemos realizar las operaciones deseadas con el objeto.

Tiene un vocabulario fácil de entender y estandarizado, pero está en inglés. Por lo que los usuarios que no sepan este idioma les resultará un poco difícil.

La información de la interfaz es comprensible y el uso es visualmente evidente, por lo tanto la claridad es buena, gracias al uso de tablas desplegables, gráficos...

En cuanto a la estética, está todo organizado en ventanas y es bastante visual.

Si queremos husmear mucho, es una herramienta para hacerlo, ya que está organizada en ventanas y cada ventana nos permite realizar un trabajo distinto a otras y comparar diferentes datos.

CRITERIOS DE CALIDAD	EVALUACIÓN
Facilidad de Auditoría	7
Seguridad	Sí
Facilidad de operar	5
Complejidad	9
Concisión	5
Consistencia	No
Auto-Documentación	7
Tolerancia de errores	6
Eficiencia en la ejecución	8
Facilidad de expansión	6
Instrumentación	6
Modularidad	8

Alta facilidad de Auditoría para poder comprobar la conformidad con los estándares.

Notoria seguridad con diferentes mecanismos que controlen o protejan los programas o datos.

Existen herramientas más sencillas a la hora de operar con ellas.

Hemos conseguido implementar todas las funciones requeridas.

No se trata de un trabajo muy compacto referido en términos de líneas de código, sino que es bastante disperso.

Poca consistencia para el uso de un diseño uniforme de técnicas de documentación a lo largo del proyecto de desarrollo de software.

Se proporciona bastante documentación significativa, lo que nos ha sido bastante útil a la hora de la implementación.

Produce un riesgo alto cuando el programa encuentra un error.

Buen rendimiento eficiente en tiempo de ejecución de un programa.

Facilidad de expansión en el grado en que se puede ampliar el diseño arquitectónico de datos.

El grado en que el programa muestra su propio funcionamiento e identifica errores que aparecen no es muy alto.

Alta independencia funcional de los componentes del programa.

7. Comparación de la implementación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

CRITERIOS	TECNOLOGÍA A	TECNOLOGÍA B	COMENTARIOS
3.1 Criterio 1: Tiempo	RapidMiner	Weka Oracle	-----
3.1.1 Criterio de tiempo de planificación	14 min	20 minutos	La herramienta de Rapid Mainer ha sido más rápida en realizar el tiempo de planificación ya que teníamos más documentación e información para planificarnos.
3.1.2 Criterio de tiempo de preparación	1h 15 min	2 horas	La herramienta de Rapid Mainer ha sido más rápida en realizar el tiempo de preparación.
3.1.3 Criterio de tiempo para la organización de los recursos	25 min	20 minutos	La herramienta de Weka Oracle ha sido más rápida en realizar el tiempo para la organización de recursos.
3.1.4 Criterio de tiempo de instalación de la tecnología a usar	10 min	8 minutos	La herramienta de Weka Oracle ha sido más rápida en realizar el tiempo de instalación de la tecnología a usar
3.1.5 Criterio de tiempo de ajuste del sistema	7 min	0 minutos	La herramienta de Weka Oracle ha sido más rápida en realizar el tiempo de ajuste en el sistema ya que su tiempo ha sido de 0 minutos.
3.1.6 Criterio de tiempo de pruebas	1h 50 min	2 horas	La herramienta de Rapid Mainer ha sido más rápida en realizar el tiempo de pruebas. (Ejecución del resultado hasta el Final)

3.1.7 Criterio de velocidad de funcionamiento del sistema	10 seg en realizar una acción o menos	10 segundos en realizar una acción	Ambas herramientas proporcionan una similitud a la hora de realizar el tiempo de funcionamiento del sistema. Es decir la ejecución del resultado hasta el Final.
3.2 Criterio 2: Funcionalidad			
3.2.1 Criterio de facilidad de uso	8	7	Sobre la facilidad de uso a la hora de realizar dicha implementación destaca rapidminer.
3.2.2 Criterio de flexibilidad	6	6	Ambas herramientas proporcionan una similitud de herramientas a la hora de adaptar diversos problemas.
3.2.3 Criterio de claridad	9	6	Rapidminer destaca por su fácil uso y entendimiento del programa.
3.2.4 Criterio de documentación	10	8	Rapidminer destaca debido a la amplia disponibilidad de manuales o guías para facilitar el uso del programa.
3.2.5 Criterio de recuperabilidad	Sí	Sí	Ambas herramientas poseen herramientas de backup.
3.2.6 Criterio de seguridad	Sí	No	Para ingresar en rapidminer se hace mediante usuario y contraseña, sin embargo, en Weka oracle no es necesario
3.3 Criterio 3: Diseño			
3.3.1 Criterio de control de usuario	sí	Sí	Ambas herramientas poseen un amplio diseño para uso libre mediante el usuario.
3.3.2 Criterio de control de Sensibilidad	sí	Sí	Ambas herramientas proporcionan cuadros de diálogos para indicar errores de usuario.

3.3.3 Criterio de control de Personalización	sí	No	Rapidminer proporciona la opción de personalizar la ventana de trabajo cuando Weka Oracle no.
3.3.4 Criterio de control de Dirección	sí	Sí	Ambas herramientas proporcionan control de dirección ya que nos deja ubicar un objeto en el sistema, ya que es intuitivo porque nos deja señalar con el mouse, y además, podemos realizar las operaciones deseadas con el objeto.
3.3.5 Criterio de control de Consistencia	sí	Sí	Ambas herramientas usan vocabulario que manejan los usuarios.
3.3.6 Criterio de control de Claridad	No	Sí	Weka Oracle utiliza una interfaz mas comprensible a la hora de realizar cualquier actividad.
3.3.7 Criterio de control de Estética	No	Sí	Weka Oracle utiliza una interfaz mas visualmente agradable.
3.3.8 Criterio de control de Indulgencia	Sí	Sí	Ambas herramientas proporcionan control de indulgencia, es decir. Si queremos husmear mucho, son herramientas para hacerlo, ya que está organizada en ventanas y cada ventana nos permite realizar un trabajo distinto a otras y comparar diferentes datos.
3.4 Criterio 4: Calidad			
3.4.1 Criterio de Facilidad de Auditoría	9	7	Ambos hacen uso de estándares, sin embargo, en Rapidminer resulta más sencilla su comprobación.
3.4.2 Criterio de Seguridad	Sí	Sí	Ambos contienen mecanismos que controlan o protegen sus datos.
3.4.3 Criterio de Facilidad de Operación	9	5	Rapidminer nos ha resultado más sencillo a la hora de operar.

3.4.4 Criterio de Completitud	9	9	En ambos hemos conseguido implementar todas las funciones requeridas.
3.4.5 Criterio de Concisión	6	5	Rapidminer es más compacto que Weka, en cuanto a líneas de código.
3.4.6 Criterio de Consistencia	Sí	No	A diferencia de Weka, Rapidminer utiliza un diseño uniforme de técnicas de documentación a lo largo del proyecto de desarrollo de software.
3.4.7 Criterio de Auto-Documentación	7	7	Ambos cuentan con una documentación de código fuente bastante significativa
3.4.8 Criterio de Tolerancia de Errores	8	6	Rapidminer es más tolerante a errores que Weka.
3.4.9 Criterio de Eficiencia en la Ejecución	10	8	Hemos quedado fascinados con el tiempo de ejecución de Rapidminer
3.4.10 Criterio de Facilidad de expansión	10	6	El diseño arquitectónico de datos de Rapidminer es fácilmente ampliable.
3.4.11 Criterio de Instrumentación	8	6	Ambos programas muestran su propio funcionamiento e identifican errores que aparecen, pero Rapidminer lo hace de una forma más eficiente que Weka.
3.4.12 Criterio de Modularidad	7	8	Los diferentes módulos de Weka son bastante independientes unos de otros.

8. Conclusiones

Como conclusión final después de haber realizado las dos implementaciones, utilizando un modelo de clasificador automático, hemos aprendido a crear un clasificador que permita diagnosticar de forma automática si un paciente tiene cáncer o no en función de los resultados de una serie de pruebas. Pero esto es solo un ejemplo. Se pueden crear clasificadores automáticos para muchas tareas.

En cuanto a las herramientas, nos ha resultado más fácil la implementación con RapidMiner, consideramos que es un programa muy intuitivo y sencillo de utilizar, siempre teniendo tutoriales a mano y guías para la elaboración de los ejercicios o casos prácticos. Te ofrece gran variedad de documentación y casos realizados. Sin embargo, oracle weka no nos ha parecido tan intuitivo y sencillo de utilizar en comparación con RapidMiner.

A la hora de importar nuestra base de datos, RapidMiner aloja más tipos de formato para cargar nuestra colección de entrenamiento. Y esto es una ventaja considerable.

Como resultado final y a modo resumen, la herramienta utilizada RapidMiner, es una muy buena ayuda para llevar a cabo un proceso de Minería de Datos, por un lado porque es muy completa en cuanto a todo lo que en este campo se necesita, y por otro porque aparte de que tiene facilidades para la enseñanza el manejo de la herramienta más a fondo, cuenta con un foro en el que se da una atención excelente y se resuelven muchas dudas que surgen a lo largo de la experiencia de realizar un proyecto de Minería de Datos.