Facultad de Ciencias Sociales y Jurídicas

Grado en Administración y Dirección de Empresas

Universidad de Jaén

Facultad de Ciencias Sociales y Jurídicas

Trabajo Fin de Grado

INTRODUCCIÓN A LA MINERÍA DE DATOS CON WEKA: APLICACIÓN A UN PROBLEMA ECONÓMICO

Alumno: Francisco Navas Moreno

Junio, 2016

Universidad de Jaén

Facultad de Ciencias Sociales y Jurídicas

Trabajo Fin de Grado

INTRODUCCIÓN A LA MINERÍA DE DATOS CON WEKA: APLICACIÓN A UN PROBLEMA ECONÓMICO

Alumno: Francisco Navas Moreno

Junio, 2016

INTRODUCCIÓN A LA MINERÍA DE DATOS CON WEKA: APLICACIÓN A UN PROBLEMA ECONÓMICO

1. INTRODUCCIÓN A LA MINERÍA DE DATOS

2. MINERÍA DE DATOS CON WEKA

ABSTRACT

Nowadays, we live in a period where enormous amounts of data in very diverse fields are being generated daily. Being able to store and access this data in a fast and clear way, along with the capability of the human being to analyze them and retrieve valuable information, has encouraged the rise of these analysis techniques known as Data Mining. Nevertheless, its potential lays on the hidden information that we could obtain from them.

The aim of this current study is to present what Data Mining is and its application to a concrete example such as the quality of wine. Afterwards, we will be able to apply this study to a local product of our land "the olive oil"

RESUMEN

Vivimos en una época donde diariamente se generan ingentes cantidades de datos en campos muy diversos. Poder almacenar y tener acceso a esos datos de forma rápida y clara, junto con la capacidad del ser humano para analizarlos y obtener información valiosa ha permitido el nacimiento de estas técnicas de análisis conocidas como Minería de Datos. No obstante, su potencial reside en la información oculta que podamos extraer de ellos.

El objetivo de este trabajo es presentar qué es la Minería de Datos y su aplicación a un ejemplo concreto como es la calidad del vino para posteriormente, en un futuro, trasladar este estudio a un producto de nuestra tierra como es el aceite de oliva.

INTRODUCCIÓN A LA MINERÍA DE DATOS

1.1.- Introducción.

Hoy en día existe una gran cantidad de información almacenadas en potentes bases de datos, pero este enorme volumen excede la capacidad que tiene el ser humano para poder sacarle beneficios directos a los mismos. No obstante, su potencial reside en la información oculta que podamos extraer de ellos. Pensemos, por ejemplo, en la poderosa información disponible en las redes sociales aportada por cada uno de los integrantes de las mismas. Según *Stephen Baker*, "*Yahoo captura una media mensual de 2.500 datos sobre cada uno de sus 250 millones de usuarios*". Pero los datos pueden extraerse de otras muchas fuentes: videocámaras de seguridad, aparatos inteligentes (smart), datos fiscales y médicos, tarjetas de crédito y de grandes superficies, wifi gratuito, parking privados, etc.

Es conveniente, en un principio, definir los conceptos con los que trabajaremos. La palabra **dato** procede del latín *datum* ("*lo que se da*") y en un sentido amplio es una información que permite el conocimiento de algo. Mientras que al conjunto de datos que están relacionados con un mismo tema lo denominaremos **información**. Por una **base de datos** entenderemos *"a una colección de información que sobre una materia se ha podido recopilar, almacenada en archivos, y que puede ser usada por distintas personas"* (http://definicion.de/datos/). Cuando a partir de los datos sea posible encontrar un modelo que represente a la situación de partida, entonces diremos que se ha generado **conocimiento**. Es decir, los datos contienen información útil que aportan conocimiento (figura 1.1).

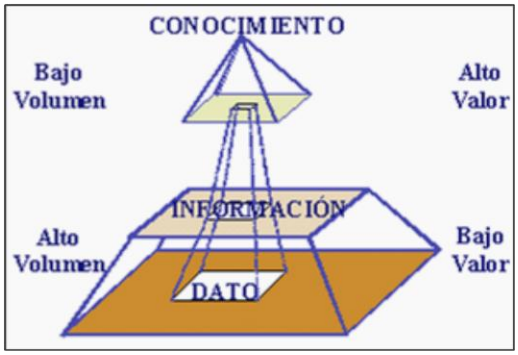


Figura 1.1. Relación entre dato, información y conocimiento (Molina, 1998).

Hablando de una manera mucho más general, el Big Data o Datos a Gran Escala, es un campo de plena actualidad, ubicado dentro de la Tecnología de la Información y de la Comunicación, que como se enuncia en la Wikipedia *"es un concepto que hace referencia al almacenamiento de grandes cantidades de datos y a los procedimientos usados para encontrar patrones repetitivos dentro de esos datos"*.

Big Data es, posiblemente, uno de los campos que más han evolucionado en los últimos años y por tanto, que más se están estudiando en estos días**: extraer información a partir de una fuente masiva de datos**. Es necesario emplear diferentes técnicas para poder obtener relaciones coherentes entre distintos atributos y con ello transformar esos datos en información necesaria para ayudar, por ejemplo, a la toma de decisiones. Para explicar todo ello hablaremos de los dos campos fundamentales: **Knowledge Discovery in Databases (KDD)** (Descubriendo

de Conocimiento en Bases de Datos) y **Minería de Datos (MD)**, este último lo estudiaremos más adelante con mayor profundidad.

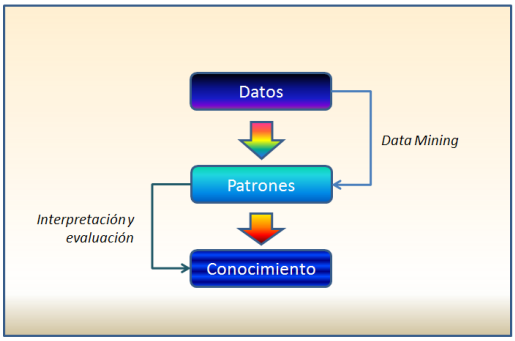


Figura 1.2. Knowledge Discovery in Databases (KDD).

El campo de estudio que afronta el trabajo que estamos abordando se conoce como KDD**.** No existe una única definición, debido a la diversidad de las técnicas utilizadas y los diferentes campos de aplicación. Según *Usama Fayyad, "KDD es el proceso no trivial de identificar patrones válidos, nuevos, potencialmente útiles y en última instancia comprensible en los datos"*.

Las propiedades del conocimiento que se extrae, son:

- **Válidos**: los patrones obtenidos deben de seguir siendo precisos para datos nuevos no solo para los que ya hayan sido usados.
- **Novedoso**s: deben aportar algo desconocido al sistema.
- **Útiles**: la información obtenida debe conducirnos a acciones que nos aporten algún beneficio.
- **Comprensible**s: la extracción de patrones debe de ser claros para el usuario, ya que de no ser así, dicha información no aportaría ningún conocimiento nuevo.

Por tanto, el KDD se puede entender como el proceso que se encarga de la extracción y preparación de los datos y de su interpretación. Resumiendo, el objetivo de este campo del conocimiento es el **encontrar patrones o relaciones entre los numerosos datos existentes, para la ayuda en la toma de decisiones**. Como podemos apreciar en la figura 1.2, el KDD tiene la capacidad de descubrir información nueva y significativa a partir del uso de los datos existentes.

De manera general, el proceso de KDD empieza identificando **qué** datos necesitamos, **dónde** podemos encontrar la información y **cómo** podemos conseguirlos. Una vez que tenemos los datos, identificando aquellos que nos sean útiles para alcanzar los objetivos que nos hemos propuestos, se preparan poniéndolos en un formato apropiado, que posteriormente se almacenan mediante un procedimiento conocido como **Data Warehouse**. Con los datos adecuados, el siguiente paso es utilizar una determinada metodología, por ejemplo, la Minería de Datos o Data Mining, para buscar, entre otras cosas, patrones de comportamiento. Durante este proceso seleccionamos herramientas y técnicas para poder obtener la información necesaria. Finalmente, se evalúan los resultados obtenidos. En la figura 1.3 se ha resumido el proceso que se ha comentado.

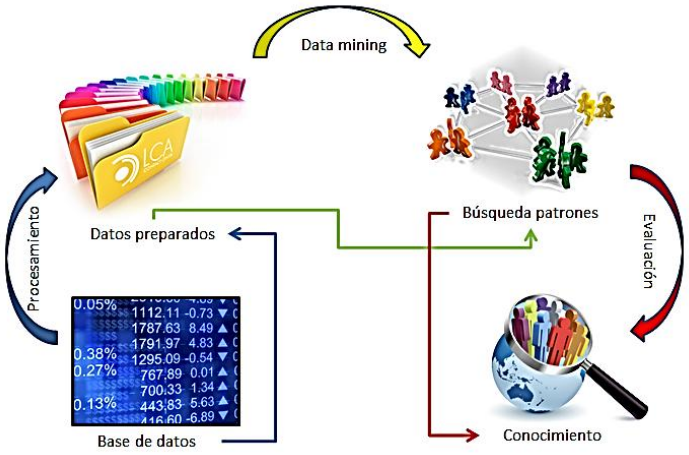


Figura 1.3. Knowledge Discovery in Databases (KDD).

El procedimiento de encontrar la información oculta deseada tiene una serie de etapas que se estructuran de la siguiente forma:

- ✓ **Creación del conjunto de datos.**
- ✓ **Limpieza y procesamiento de datos**: se basa en recoger los datos necesarios y elegir la estrategia a la hora de manejar los datos.
- ✓ **Reducción de datos y proyección**: encontrar las características significativas de los datos.
- ✓ **Elegir la tarea de la minería de datos**: seleccionar el objetivo de KDD; agrupamiento, regresión, clasificación...

- ✓ **Elección del algoritmo o los algoritmos de la MD**: método utilizado para la búsqueda de patrones.
- ✓ **Interpretación de los patrones encontrados**: ver si la información extraída nos resulta útil.

Estos pasos se deben de seguir para extraer la información del conjunto de datos que tenemos, pero, en nuestro trabajo, nos centraremos en una etapa concreta que es la MD, ya que ésta es la parte fundamental para poder encontrar la información que se necesita.

1.2.- Minería de Datos.

1.2.1.- El nacimiento de la Minería de Datos. La MD ha ido evolucionando durante el siglo XX, hasta hacerse muy importante en los últimos años. Vivimos en una época donde diariamente se generan ingentes cantidades de datos en campos muy diversos. Poder almacenar y tener acceso a esos datos de forma rápida y clara, junto con la capacidad del ser humano para analizarlos y obtener información valiosa ha permitido el nacimiento de estas técnicas de análisis conocidas como MD.

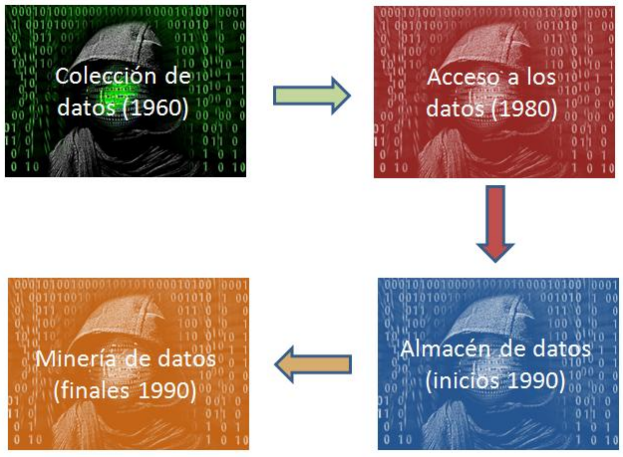


Figura 1.4. Evolución histórica

Es evidente la vinculación de la Estadística con esta rama del conocimiento. Por lo tanto, no sorprende que en un principio fuese la ciencia encargada de extraer información básica de un conjunto de datos experimentales. Con los avances tan impresionantes de la Informática en los años sesenta del siglo pasado, los investigadores empezaron a utilizar en Inteligencia Artificial estas herramientas, generando lo que se conoce como **Machine Learning** o **Aprendizaje**

Automático en Máquinas. Es un área interdisciplinar compuesta por científicos procedentes de la inteligencia artificial, estadísticos, biólogos, matemáticos y neurocientíficos.

De una manera más concreta, en los años cuarenta se publicaron los primeros trabajos interesados en reproducir el comportamiento del cerebro a través de modelos matemáticos que se conocen con el nombre de **redes neuronales.** Estos algoritmos **e**stán basados en el funcionamiento del sistema nervioso de los animales y fueron publicados en 1943 por los neurólogos *Warren McCulloch* y *Walter Pitts*. Estos modelos se diferenciaban del resto porque tenían la ventaja de poder imitar el proceso de aprendizaje de nuestro cerebro. De hecho, en la actualidad Google está haciendo un uso intenso de las redes neuronales.

Fuente imagen: http://comunicacion3unlz.com.ar/conocimiento-colaborativo/mineria-de-datos/

Al inicio de los años sesenta algunos estadísticos empezaron a usar ciertas técnicas propias de la MD, conocidas como Data Fishing, o Data Archaeology, cuya idea era encontrar semejanzas sin platearse hipótesis previas. En estos años los investigadores estaban muy interesados en la forma en el que queda representado el conocimiento, surgiendo, de la inteligencia artificial, lo que hoy conocemos como árboles de decisión, expresiones lógicas, etc., en lugar de los clásicos métodos estadísticos.

A finales de los años ochenta, empiezan a aparecer empresas dedicadas a la MD, aunque por esa época, y debido a su dificultad e inexistencia de software específico, solo existían dos empresas dotadas de esa tecnología. Todo esto contrasta con el aumento significativo ocurrido a principios del siglo XX, donde estas compañías se hacen más numerosas. Como personajes pioneros en este periodo podemos citar:

- ***Rakesh Agrawal*** trabajó en un principio en los laboratorios de IBM (Almaden Research Center). Entre 1983 y 1989 formó parte del grupo de investigación de Murray Hill en el Bell Laboratories, y tras distintas estancias en otras empresas, se incorporó definitivamente como informático en 2006 a la plantilla de Microsoft. Es una de las personas que más han influido en el

desarrollo de los aspectos fundamentales de la MD y de conceptos claves en la privacidad de datos. Por ejemplo, Intelligent Miner que es el producto específico de IBM en MD fue creado a partir de sus primeros trabajos. En su haber cuenta con otros productos, prototipos y aplicaciones comerciales y académicas como: DB2, incluyendo Minería Extender, DB2 OLAP Server y WebSphere Commerce Server. Como curiosidad diremos que en 2003 la revista Scientific American lo incorporó a la lista de los 50 científicos más importantes e influyentes de la actualidad. Para una información más completa puede visitarse la página de su fundación (http://rakeshagrawal.org/)

- Otro de los pioneros en el uso de la MD fue el ingeniero aeronáutico **Giovanni "Gio" *Corrado Melchiore Wiederhold,*** nacido en 1936 en Varese, Italia. Tras trabajar en muy diversas instituciones (OTAN, IBM, Universidad de California), finalmente ha desarrollado la mayor parte de su carrera investigadora en la Universidad de Stanford, donde ingresó en 1976. En la actualidad es profesor emérito de esta institución. Sus estudios están dirigidos al diseño de grandes sistemas de gestión de bases de datos, y a la protección de su contenido utilizando técnicas basadas en el conocimiento. Todas estas técnicas las puso en práctica en la Escuela de Medicina de la Universidad de Stanford.

- ***Gregory Piatetsky-Shapiro*** es hijo de *Ilya Piatetski-Shapiro* (1929, Moscú - 2009, Tel-Aviv), famoso matemático que realizó importantes aportaciones al campo de la teoría analítica de números y a la geometría algebraica. Nació en Moscú, donde empezó a estudiar en la Escuela Superior de Matemáticas, pero su interés se centró en el estudio y uso de los ordenadores. En 1974 *Piatetsky* emigró a Israel con su madre donde estudió matemáticas en la Universidad de Tel Aviv, siendo con la edad de 16 años el estudiante más joven. Posteriormente se trasladó a Estados Unidos doctorándose en 1984 en la Universidad de New York con un tema de aplicación de aprendizaje automático a las Bases de Datos. En 1989 organizó el primer taller de descubrimiento de conocimiento en los datos (KDD-89) celebrado en Detroit. Actualmente es presidente de KDnuggets empresa que creó años antes de que

Sergey Brin y Larry Page fundaran Google. Se define como un científico de datos interesado en la comprensión de cómo funciona el mundo. En http://www.analyticsvidhya.com/blog/2015/10/interview-data-scientist-gregory-piatetsky-shapiro-president-kdnuggets/ puede verse una interesante entrevista.

Resumiendo, hoy en día cada acción que realizamos en la que interactuamos con una persona, o con una empresa, queda registrado informáticamente y puede constar como dato almacenado. Existen inagotables fuentes de datos como pueden ser, entre otros: internet, las transacciones de las tarjetas de crédito de un banco, las operaciones de compra-venta, los resultados experimentales en ciencias biológicas o en medicina, los informes gubernamentales, etc. Es usual que el estudio de estos datos sea muy difícil debido al elevado número de variables que intervienen (atributos) y sus propiedades interesantes se diluyan y confundan al investigador.

Por tanto, la mayoría de las decisiones importantes que se toman, se basan en observaciones que han sido almacenadas en una base de datos, por lo que la MD se ha desarrollado como una herramienta imprescindible en el mundo en el que vivimos.

1.2.2.- Definiciones de Minería de Datos. Debido a lo novedoso de la herramienta, no existe una única definición en la que coincidan un grupo importante de investigadores, pero puede resumirse en las siguientes:

- ✓ MD es el proceso de encontrar correlaciones o patrones entre múltiples atributos en grandes bases de datos.
- ✓ *"MD es la extracción de información implícita, desconocida o previamente ignorada, que puede ser útil de un conjunto de datos"* (Vilches González, et al, 2007).
- ✓ *"MD lies at the interface of statics, database, pattern recognition and machine learning"* (Riganello, et al, 2010).
- ✓ *"El proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos"* (Hernández-Orallo, 2007).
- ✓ "*MD* es *el proceso de planteamiento de distintas consultas y extracción de información útil, patrones y tendencias previamente desconocidas desde grandes cantidades de datos posiblemente almacenados en bases de datos*".

Entonces, ¿qué se puede entender como la MD? ¿Cómo podemos definirla? Se puede resumir como el proceso que intenta descubrir patrones en grandes volúmenes de datos para extraer una información útil. De manera parecida a como la minería tradicional intenta recuperar el mineral valioso de un gran volumen de tierra. La idea clara que debemos obtener de la MD es la de que se intenta convertir los datos en conocimiento, para ello se hace necesario encontrar modelos o relaciones a partir de datos que ya están almacenados para que ayuden a tomar las decisiones más seguras. Se presentan dos grandes retos:

1. Trabajar con grandes volúmenes de datos, presentándose los problemas propios de los sistemas de información como son: la ausencia de datos, su volatilidad, intratabilidad, etc.
2. Usar las técnicas de análisis adecuadas, para poder analizar y extraer informaciones útiles de los mismos.

Una vez introducida la MD, y una vez comentado cómo ha ido evolucionando a lo largo de los últimos años, debemos insistir en que su aparición no es debida a la evolución de la tecnología, sino que nace a partir de la creación de nuevas necesidades. Estas necesidades se crean como consecuencia del aumento de la cantidad y variedad de información que se encuentran registradas en las bases de datos, especialmente en las últimas décadas.

La situación de convertir los datos en conocimiento, analizar e interpretar datos, se puede llevar a cabo de forma manual. Con el paso de los años, concretamente en las últimas décadas, esta forma de interpretar y analizar datos de forma manual se vuelve más compleja ya que el gran aumento del volumen de datos de los últimos años hacen que se desborde la capacidad humana para obtener información útil de las bases de datos si no es con ayuda de la tecnología.

Un ejemplo básico con el qué se puede explicar cómo obtener información útil es el uso de las tarjetas de los clientes de los Supermercados Día.

Idcesta	Huevos	Aceite	Pañales	Vino	Leche	Mantequilla	Salmón	Lechugas	...
1	sí	no	no	sí	no	sí	sí	sí	...
2	no	sí	no	no	sí	no	no	sí	...
3	no	no	sí	no	sí	no	no	no	...
4	no	sí	sí	no	sí	no	no	no	...
5	sí	sí	no	no	no	sí	no	sí	...
6	sí	no	no	sí	sí	sí	sí	no	...
7	no	no	no	no	no	no	no	no	...
8	sí	sí	sí	sí	sí	sí	sí	no	...
...	...	...	...	...	...	...	...	...	...

Fuente: Introducción a la Minería de Datos (José Hernández Orallo, 2007)

El funcionamiento de estas tarjetas es simple, cada cliente obtiene un descuento al pasar por caja y abonar su compra. Para el supermercado, la función de estas tarjetas es la de registrar la cesta de la compra de cada cliente por cada paso por caja, de esta manera, se puede mejorar el servicio que se ofrece.

Analizando estos patrones de compra se puede obtener información sobre las relaciones que hay entre productos. Por ejemplo:

1. Se sabe que todas las veces que se compra pañales también se suele comprar leche.
2. La mayoría de las veces que se compran huevos también se adquiere aceite.
3. Con la información recopilada por el uso de dichas tarjetas, Supermercados Día obtiene información útil con el fin de reubicar los productos que se suelen comprar juntos, o colocar los productos nuevos al lado de los que se venden más, etc.

Es importante aclarar que la MD es una parte del proceso del descubrimiento de conocimiento KDD. Algunas personas consideran que son lo mismo pero lo cierto es que son conceptos diferentes. La MD la usan los estadísticos, informáticos y analistas de datos mientras que KDD, se refiere a un proceso que tiene distintas fases mientras que la MD se considera una de esas fases, y se utiliza más en Inteligencia Artificial.

1.2.3.- Elementos fundamentales en la Minería de Datos. Las técnicas que se consideran más importantes en la MD son los métodos estadísticos y el aprendizaje automático. La primera de ella produce paquetes estadísticos que sirven para computar sumas, hacer promedios y distribuciones que se van integrando a las bases de datos que vamos a estudiar. Por otra parte, el aprendizaje automático se basa en obtener modelos de datos y reglas de aprendizaje que se obtienen a través del uso de la Estadística. Aunque estas se considerarían como las más importantes, hay otras tecnologías como las técnicas de visualización, las técnicas de procesamiento paralelo y los sistemas de apoyo a la toma de decisiones. La primera facilita la presentación de datos a la minería; la segunda, ayuda a mejorar su rendimiento; y la tercera, rechaza los resultados que no nos sirven y proporcionan los que son esenciales para llevar a cabo nuestros objetivos.

1.2.4.- Tareas en la aplicación de la Minería de Datos. Usualmente, la MD utiliza cuatro clases de tareas: clasificación, agrupamiento (clustering), regresión, y reglas de asociación. Para

ello utiliza distintas técnicas como los modelos estadísticos, los algoritmos matemáticos y los algoritmos de aprendizaje automático que mejoran su rendimiento a través de la experiencia, como las redes neuronales o los árboles de decisión. A continuación comentaremos las tareas usuales en la MD.

1.2.4.1.- Clasificación y predicción. Es la más interesante de las tareas predictivas y consiste en localizar las relaciones existentes entre todos los datos (o parte de ellos), con la intención de que se puedan utilizar estos patrones en futuras predicciones. Su aplicación es usual en el estudio de la concesión o no de hipotecas en un banco, en la obtención del tipo de clientes según el modo en que utilicen la tarjeta bancaria, o en la detección de reconocimiento de caras en un conjunto de imágenes. Para ello, se utiliza un conjunto de datos conocidos para extraer las reglas de clasificación y posteriormente se aplican estas reglas al conjunto de los datos cuyo comportamiento se quiere predecir. Los algoritmos más utilizados en la clasificación son los árboles de decisión, la clasificación bayesiana, las redes neuronales y los clasificadores basados en reglas (por ejemplo, clasificación por la regla de los k-vecinos más cercanos).

1.2.4.2.- Agrupamiento. Es una tarea descriptiva, y consiste en obtener grupos a partir de los datos registrados. En este caso se habla de grupos y no de clases como en la clasificación. El objetivo de la tarea es agrupar los datos en grupos de tal manera que los elementos de cada grupo sean muy similares entre sí y que al mismo tiempo sean muy diferentes a los del resto de otros grupos.

1.2.4.3.- Regresión. Pertenece al grupo de las tareas predictivas**.** Ahora, el objetivo es el de utilizar algoritmos con la intención de identificar las funciones que modelen con el menor error posible a la base de datos, de tal manera que la diferencia entre el valor predicho y el real sea mínimo.

1.2.4.4.- Reglas de asociación. Es otra de las tareas predictivas, que proporcionan algoritmos con la intención de encontrar relaciones relevantes entre las variables (atributos).

En la figura 1.5 aparece un esquema del tipo de técnica utilizadas en la MD y algunos de sus algoritmos.

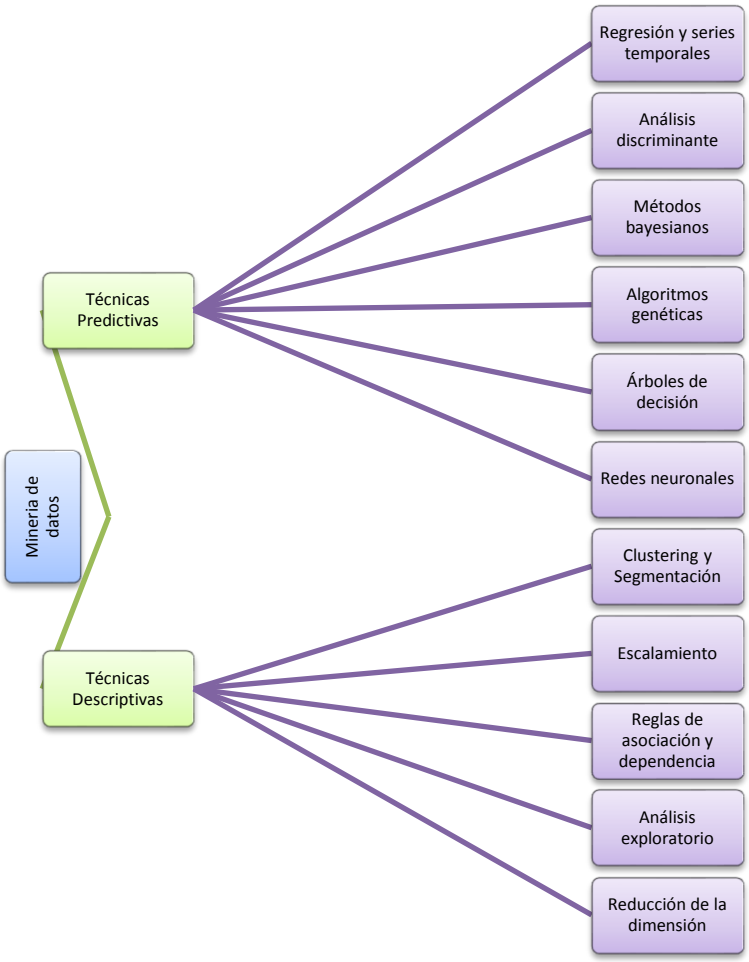


Figura 1.5. Técnicas y algoritmos en la Minería de Datos.

1.3.- Aplicaciones.

En la actualidad la MD se aplica a casi todas las ramas del conocimiento debido a que la tecnología que se utiliza es relativamente barata y además los recursos informáticos disponibles cada vez son más numerosos y sofisticados. El uso de las técnicas de MD en las actividades del día a día de una empresa se está convirtiendo en algo habitual y se está ampliando sobre todo para ayudarlas en su toma de decisiones. Entre ellas, las empresas de publicidad y distribución son las que más se han beneficiado ya que le ha permitido reducir sus costes y aumentar la sus servicios. Éstas, no son las únicas áreas a las que se puede aplicar, a continuación comentaremos algunos ejemplos sin ánimo de ser exhaustivos.

1.3.1.- Empresariales. Con carácter general, el objetivo de la MD en el mundo de la empresa es el de encontrar patrones de compras de los clientes, con la intención de promover campañas publicitarias personalizadas y de esta manera incrementar las ventas.

Una de las aplicaciones más interesante de la MD en el ámbito empresarial es la detección de fraudes en las tarjetas de crédito. Se ha estimado que a principios del año 2001 las instituciones financieras a escala mundial perdieron unos 2000 millones de dólares, debido al uso fraudulento de las tarjetas de crédito. El sistema, desde hace 15 años, encargado de analizar las transacciones y a los propietarios de las tarjetas se conoce con el nombre de *Falcon Fraud Manager* y ha supuesto que las entidades financieras de todo el mundo se hallan ahorrado más de 450 millones de dólares al años. Otra de las aplicaciones importantes de la MD en el mundo empresarial es el estudio de la migración de clientes entre distintas compañías de un mismo sector, por ejemplo el de telefonía. En este caso se analizan, entre otros, factores como el perfil de clientes que se dan de baja, o el comportamiento de los nuevos clientes. Como curiosidad citaremos que a raíz de las conclusiones de estos estudios, se detectó que los clientes que se daban de baja recibían pocas promociones y tenían más incidencias respecto de la media. Por ello, se recomendó a la compañía hacer un estudio sobre sus ofertas y analizar las incidencias recibidas por estos tipos de clientes.

1.3.2.- Internet. Otra de las aplicaciones de la MD se basa en aplicar estas técnicas a documentos y servicios Web, en este caso la MD recibe el nombre de Web Mining (WM). Cada vez que un usuario visita una página web deja todo tipo de "huellas" (direcciones, navegador, cookies...) que los servidores almacenan en una base de datos. A partir de esa base de datos, la WM analiza y procesa todos esos datos para poder producir una información significativa de cómo es la navegación de un determinado cliente, ofreciéndole en la próxima visita a una página una publicidad personalizada. Por ejemplo, Google utiliza la MD para conocer el ánimo de sus 20000 trabajadores y de esta manera detectar por adelantado cuáles de sus cargos directivos (ejecutivos, ingenieros, etc.) desean marcharse a otra empresa. Como señala el profesor de la UPC Lluis Belanche, "*se trata de crear un modelo predictivo del comportamiento"* con aplicaciones inmediatas en el campo de los recursos humanos, la medicina y en el terrorismo.

1.3.3.- Deporte. En los equipos de fútbol y baloncesto ha empezado ser frecuente el uso de la MD. En el caso del fútbol, el A.C. Milán posee un sistema inteligente para prevenir lesiones y optimizar el rendimiento del jugador basado en un sistema de redes neuronales. Este sistema fue

creado por *Computer Associates International* y se alimenta por los datos que proporciona cada jugador debido a los veinticuatro sensores que tiene conectados en el cuerpo, recogiendo datos sobre su alimentación, rendimiento y su respuesta a estímulos externos. Este sistema tiene cinco mil casos registrados que permiten predecir alguna lesión evitando así que el club fiche a jugadores que tengan una alta probabilidad de lesión y así ahorrar dinero.

En el caso del baloncesto, los equipos de la NBA también utilizan MD para ayudar a los entrenadores. IBM desarrolló un programa que registra las estadísticas de cada partido: pases, canastas, rebotes... El objetivo es ayudar a los entrenadores a analizar cosas que no pudo ver durante el partido o en la repetición del mismo. Por ejemplo, el entrenador de los *New York Knicks,* a través de este programa, observó que cada vez que jugaba contra los *Chicago Bulls* reaccionaban rápidamente al doble marcaje y que podían tapar de forma rápida al jugador libre de *los Knicks* antes de que efectuara su tiro. De esta forma creo estrategias alternativas para evitar el doble marcaje.

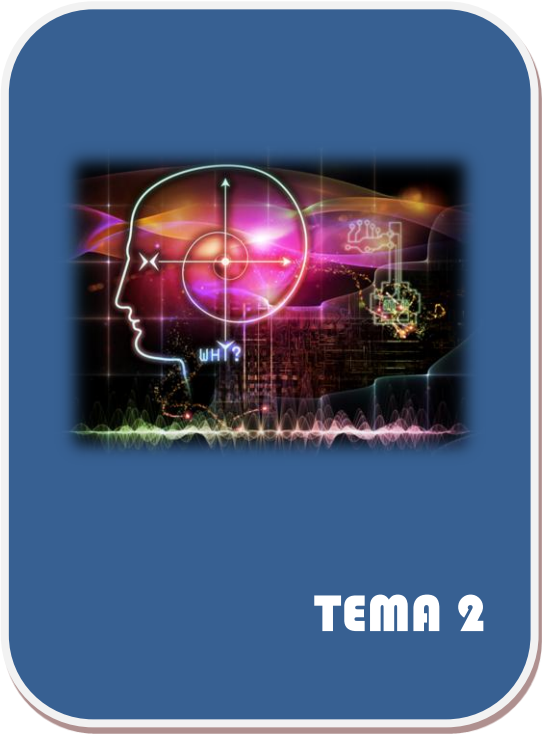
1.3.4.- Software. Tradicionalmente el método utilizado para interpretar los datos es manualmente. Sin embargo, en la actualidad es imposible hacerlo de esta manera debido al elevado volumen de información disponible, ya que ésta crece de forma exponencial y es necesario recurrir a las nuevas tecnologías. Existe un gran número de software disponible para trabajar en MD, entre los más conocidos, gratuitos y de código abierto, se encuentran:

- **Orange:** es un programa que utiliza el lenguaje C++ y Python, desarrollado en la Universidad de Ljubljana (Eslovenia). Cuenta con la posibilidad de un preprocesamiento muy interesante de los datos, con los algoritmos más usuales de MD, y una interface gráfica muy completa. Se distribuye bajo licencia GPL.

- **RapidMiner**: es un software de MD ampliamente utilizado tanto para investigación como para el mundo empresarial y en la creación de prototipos. Dispone de más de 500 rutinas para los principales procedimientos de aprendizaje. La herramienta permite: preparar los datos, crear modelos e introducir los resultados en los procesos de negocios de una forma rápida. Es interesante hacer constar que el programa ofrece la posibilidad de poder ejecutar la mayoría de los algoritmos utilizados en WEKA.

- **JHepWork**: es otro de los programas de MD diseñado para estudiantes, científicos e ingenieros de código abierto para el análisis de datos. Se creó para que su interfaz sea fácil y sencilla de utilizar y para que esta herramienta fuera competitiva con los demás programas comerciales. Tiene su origen en 1990 en el estudio de la física de alta energía y está basado en programación Java y en el lenguaje Python.

- **WEKA**: será el software que utilizaremos durante este trabajo, por lo que en el próximo tema hablaremos de manera más detallada de todas las utilidades que podemos obtener de él. El programa está basado en un conjunto de librerías Java bajo licencia GPL, y ha sido desarrollado en la Universidad de Waikato, de ahí el nombre de WEKA (*Waikato Environment for Knowledge Analysis).* Está orientado a la extracción de conocimiento a través de bases de datos con gran cantidad de información, contiene una gran colección de algoritmos y herramientas para analizar los datos junto con una interfaz sencilla, que hace que el usuario pueda usar este software de manera muy simple.



MINERÍA DE DATOS CON WEKA

2.1.- Introducción al programa WEKA.

2.1.1. Aspectos generales. El software WEKA para el estudio de la MD es gratuito y puede descargarse de la siguiente dirección: http://www.cs.waikato.ac.nz/ml/WEKA/downloading.html

Una vez instalada y ejecutada la versión 3.6.13, aparecerá la interface principal, que se corresponde con la ventana de la figura 2.1, que ofrece cuatro posibilidades de trabajo: **Explorer, Experimenter, KnowledgeFlow y Simple CLI**.

¿Para qué sirven cada una de estas opciones? Pues bien, hablaremos un poco de todas ellas, aunque nos centraremos en `Explorer`, que es el modo más usado ya que permite acceder a la mayoría de funcionalidades que tiene WEKA y realizar operaciones sobre un archivo de datos.

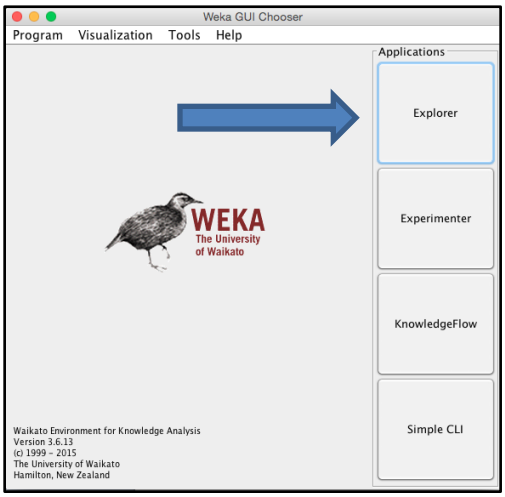


Figura 2.1. Interface principal de WEKA (versión 3.6.13).

- **Explorer:** Es el interface visual de WEKA para trabajar de manera gráfica de una manera sencilla. Éste modo permite procesar, clasificar, asociar y visualizar datos de una manera fácil e intuitiva sobre un sólo archivo de datos.
- **Experimenter:** es un modo útil para aplicar uno o varios métodos de clasificación de manera automática. Con esta ventana se facilita la realización de experimentos a gran escala.
- **KnowledgeFlow:** es el interface gráfico, y es utilizado para desarrollar proyectos a través de flujos de información.
- **Simple CLI**: se le conoce como interface línea de comandos, y se usa para llamar directamente a los paquetes de Java que WEKA incorpora.

Una vez abierta la ventana **Explorer**, que será la más útil para nosotros, se pueden observar (figura 2.2) seis pestañas que nos permitirán realizar las siguientes tareas:

- **Preproces:** es el primer paso para poder empezar a trabajar, y definir el origen de los datos. Las herramientas de preprocesamiento en WEKA se llaman filtros, y contiene, entre otros, filtros para la discretización, normalización, reemplazamiento y combinación de atributos. El tipo de filtros más utilizados son los no supervisados sobre los atributos. Aquellos que son independientes de los algoritmos aplicados.

Es evidente que lo primero será cargar el conjunto de datos, y esto puede hacerse de cuatro formas diferentes:

- abriendo un archivo a través de **Open File**
- abriendo un archivo a través de una dirección de internet **Open URL**
- abriendo una base de datos con **Open DB**
- generarlos por medio de la pestaña **Generate**.

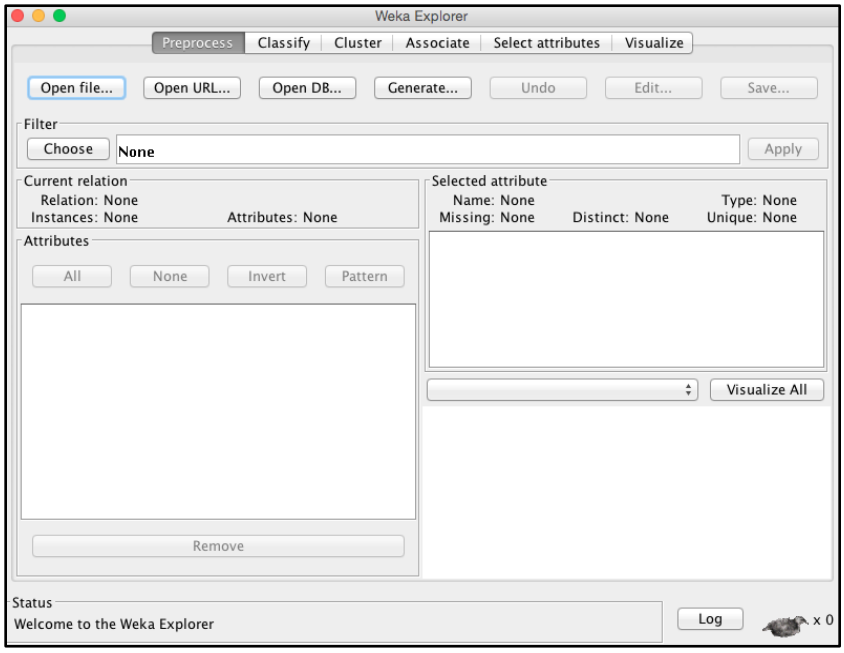


Figura 2.2. Ventana correspondiente al Explorer de WEKA.

- **Classify:** pulsando en esta segunda pestaña, entramos en el modo clasificación, conocida en algunas ocasiones como aprendizaje supervisado. Con esta opción, podemos clasificar los datos haciendo uso de técnicas, entre otras, de clasificación y regresión.
- **Cluster:** esta opción dispone de distintos algoritmos con los que se puedan agrupar los datos en base a uno o varios criterios.
- **Associate:** en esta cuarta pestaña se pueden encontrar reglas de asociaciones entre los datos. Se considera la opción más fácil de utilizar de las seis, y consiste en elegir el método deseado para encontrar asociaciones entre los datos, así como la posibilidad de configurarlo.
- **Select atributes:** con esta pestaña podremos acceder a la selección de atributos, cuyo objetivo es identificar qué conjunto de datos poseen atributos similares con el objetivo, entre otros, de reducir su número. Dentro de esta opción tenemos que seguir dos pasos;

primero, seleccionar el método para la evaluación de atributos a través de `Atribute Evaluator`, que sirve para asignar a cada atributo un peso específico. El segundo y último paso será elegir el método de búsqueda.

- **Visualize:** este apartado ofrece la posibilidad de ver de forma gráfica cómo se distribuyen todos los atributos que representan los posibles pares de combinaciones, con el objetivo de extraer información por medio de técnicas visuales.

2.1.2. Ejemplo motivador. Objetivo Una vez descritas las opciones básicas del programa, pasaremos a la realización de un ejemplo concreto para ver su funcionamiento. El **objetivo** que se pretende es encontrar patrones de comportamientos para un conjunto de datos extraídos de un equipo de futbol (Anexo I), en nuestro caso el Real Madrid, de una base de datos disponible en:

http://www.ceroacero.es/team_season.php?comp_id=5&epoca_id=0&ond=&id=39_o=

TEMPORADA		PUNTOS	POSICIÓN	PARTIDOS JUGADOS	GANADOS	PERDIDOS	EMPATADOS	GOLES FAVOR	GOLES CONTRA	CAPACIDAD ESTADIO
1928	1929	34	2	18	11	6	1	40	27	15.000
1929	1930	24	5	18	7	8	3	45	42	15.000
1930	1931	25	6	18	7	7	4	24	27	15.000
1931	1932	38	1	18	10	0	8	37	15	15.000
1932	1933	41	1	18	13	3	2	49	17	15.000
1933	1934	32	2	18	10	6	2	41	29	15.000
1934	1935	49	2	22	16	5	1	61	34	15.000
1935	1936	42	2	22	13	6	3	62	35	15.000
1939	1940	37	4	22	12	9	1	47	35	15.000
1940	1941	35	6	22	11	9	2	51	38	15.000
1941	1942	47	2	26	14	7	5	65	43	15.000
1942	1943	35	10	26	10	11	5	52	50	15.000

Tabla 2.1. Datos temporada 1928/29 al 1942/43.

Para ser más preciso, nos proponemos saber si podemos relacionar el atributo `posición` en que terminó la liga, con el resto de los atributos: como `número de puntos obtenidos`, `partidos ganados`, `partidos perdidos`, y `empatados`, y además con `goles a favor`, `goles en contra` y `capacidad del estadio`. Se dispone en total de ocho atributos y 81 instancias que van de la temporada 1928/29 hasta la temporada 2011/12. En la Tabla 2.1 aparecen sólo 12 de estas instancias.

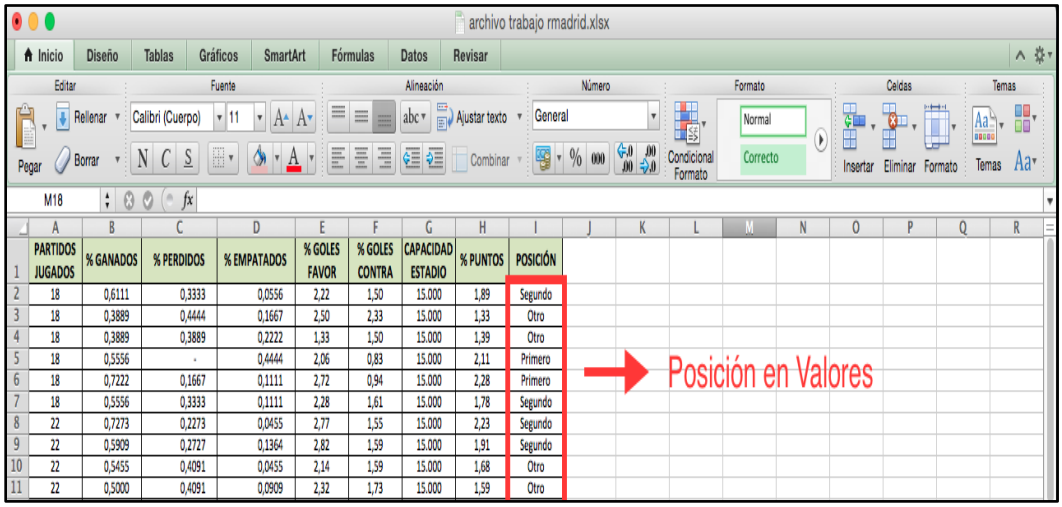
2.1.3.- Manejo de archivos en WEKA. El software trabaja con un formato de archivo que se denomina arff (Atributte Relation File Format), y que consta de tres partes:

1. **Cabecera**: donde se define el nombre del archivo a través de la expresión `@relation <nombre-de-la-relación>`.
2. **Declaración de atributos:** se corresponde con el segundo bloque y es el lugar donde se definen los atributos que vamos a estudiar, así como su tipo, por medio de la expresión `@attribute<nombre-del-atributo><tipo>`. Los tipos de atributos con los que WEKA trabaja son:
    - **Números reales**: NUMERIC
    - **Números enteros**: INTEGER
    - **Fechas**: cuyo formato es:
        - Día: dd
        - Mes: MM
        - Año: yyyy
        - Horas: HH
        - Minutos: mm
        - Segundos: ss
    - **Cadenas de texto**: STRING
    - **Enumerados**: Se expresa entre llaves y separados por comas los distintos valores del atributo. Por ejemplo:
      `@attribute posición {Primero, Segundo, Tercero, Otro}`
3. **Sección de datos:** es la parte final de archivo, se inicia con `@data`, y es el lugar donde insertamos los datos que componen nuestra base de datos. Los atributos deben estar separados por comas y con saltos de línea cada relación (instancias).

Existe la posibilidad de pasar de un archivo en Excell a un archivo con formato arff, para lo cual será necesario realizar los siguientes pasos:

- Desde la base de datos, en Excell, que aparece en la tabla 2.1 (Anexo I), tendremos que eliminar las columnas cuyos atributos son irrelevantes para nuestro estudio, como son la inversión, el presupuesto y la temporada.
- Puesto que cada una de las temporadas el equipo juega un número distintos de partidos, para poder comparar unos datos con otros debemos pasar los distintos atributos a porcentajes, es decir, dividiremos cada atributo entre el número de partidos jugados.

- Como todos los datos son homogéneos, entonces borramos las columnas que no necesitamos y alteramos (por comodidad) su orden, con objeto de facilitar luego las relaciones de los atributos con la posición obtenida a final de temporada. Quedando el archivo tal y como aparece en la Tabla 2.2



PARTIDOS JUGADOS	% GANADOS	% PERDIDOS	% EMPATADOS	% GOLES FAVOR	% GOLES CONTRA	CAPACIDAD ESTADIO	% PUNTOS	POSICIÓN
18	0,6111	0,3333	0,0556	2,22	1,50	15.000	1,89	Segundo
18	0,3889	0,4444	0,1667	2,50	2,33	15.000	1,33	Otro
18	0,3889	0,3889	0,2222	1,33	1,50	15.000	1,39	Otro
18	0,5556	-	0,4444	2,06	0,83	15.000	2,11	Primero
18	0,7222	0,1667	0,1111	2,72	0,94	15.000	2,28	Primero
18	0,5556	0,3333	0,1111	2,28	1,61	15.000	1,78	Segundo
22	0,7273	0,2273	0,0455	2,77	1,55	15.000	2,23	Segundo
22	0,5909	0,2727	0,1364	2,82	1,59	15.000	1,91	Segundo
22	0,5455	0,4091	0,0455	2,14	1,59	15.000	1,68	Otro
22	0,5000	0,4091	0,0909	2,32	1,73	15.000	1,59	Otro

Tabla 2.2. Archivo Excell temporada 1928/29 al 1942/43.

- A continuación, grabamos el archivo de Excel en formato *csv delimitado por comas*, para poder manipularlo antes de abrirlo con WEKA (Figura 2.3).

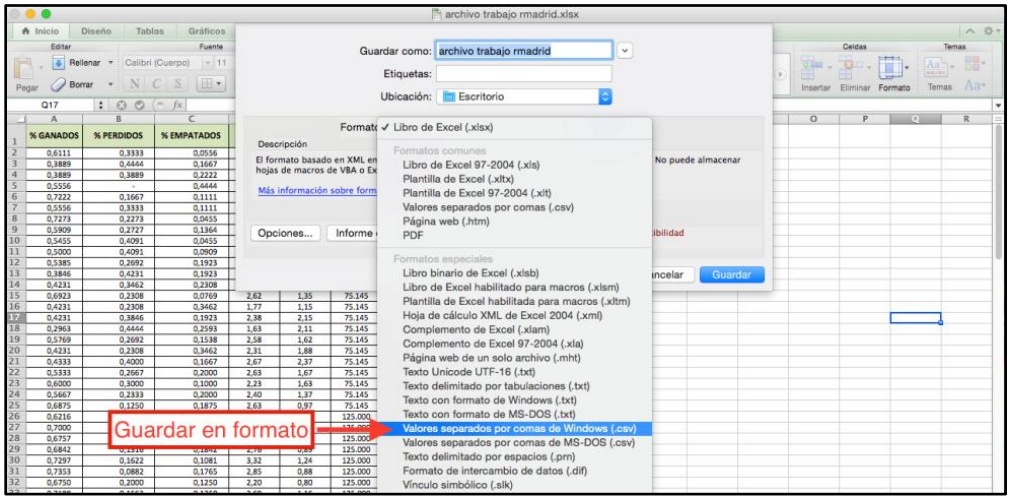


Figura 2.3. Formato grabación archivo

- Listo el fichero en formato csv, lo abrimos por medio del bloc de notas (botón derecho/abrir con/bloc de notas) y realizamos los siguientes cambios: reemplazamos las comas (,) por puntos (.) y los puntos y comas (;) por comas (,)

- Finalmente, escribimos en el archivo la cabecera, las características de los atributos y los datos que queremos estudiar (figura 2.4), y grabamos el archivo con la extensión arff

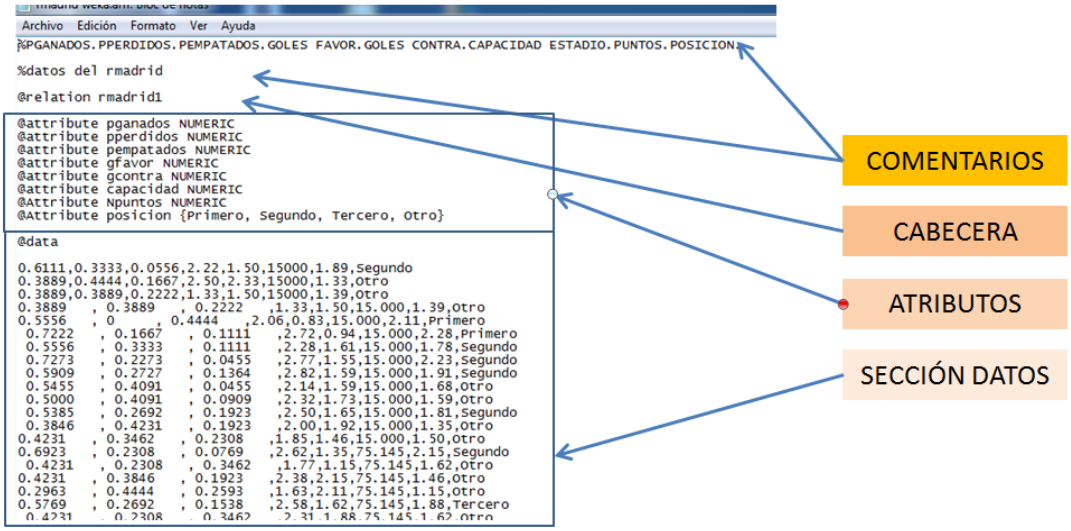


Figura 2.4. Archivo con formato arff para WEKA.

2.2. Análisis de los datos con WEKA.

Una vez que ya tenemos nuestro archivo en formato arff pasaremos a la fase de análisis, para ello abriremos el programa y pulsaremos en la pestaña de `Explorer` (explorar) para poder cargar nuestro archivo, que contiene los ocho atributos y las 82 instancias. Abierto el archivo (botón Open file) desde WEKA aparece en pantalla un resumen de los datos (Figura 2.5)

Pulsando en cada atributo podemos obtener información sobre cada uno de ellos, como por ejemplo, acerca de qué tipo se trata, si es nominal o numérico, el valor máximo o mínimo que pueden obtener los atributos numéricos…,etc. Observemos, como aspecto muy importante, que en la parte inferior derecha (propiedades de los atributos) puede verse representado geométricamente un histograma con los valores que toma el atributo seleccionado. Existe también la posibilidad de ver un histograma con todos los atributos a la vez.

2.2.1. Preprocesamiento de los datos. El primer paso para un análisis de datos con WEKA es el preprocesamiento de los mismos en la pestaña `preprocess` (Figura 2.5). Esta operación se lleva a cabo mediante el uso de filtros, que pueden ser aplicados a los atributos o a las instancias. En general, el tipo de filtro es no supervisado, esto es, el resultado obtenido es independiente del tipo de algoritmos que se utilice a posteriori.

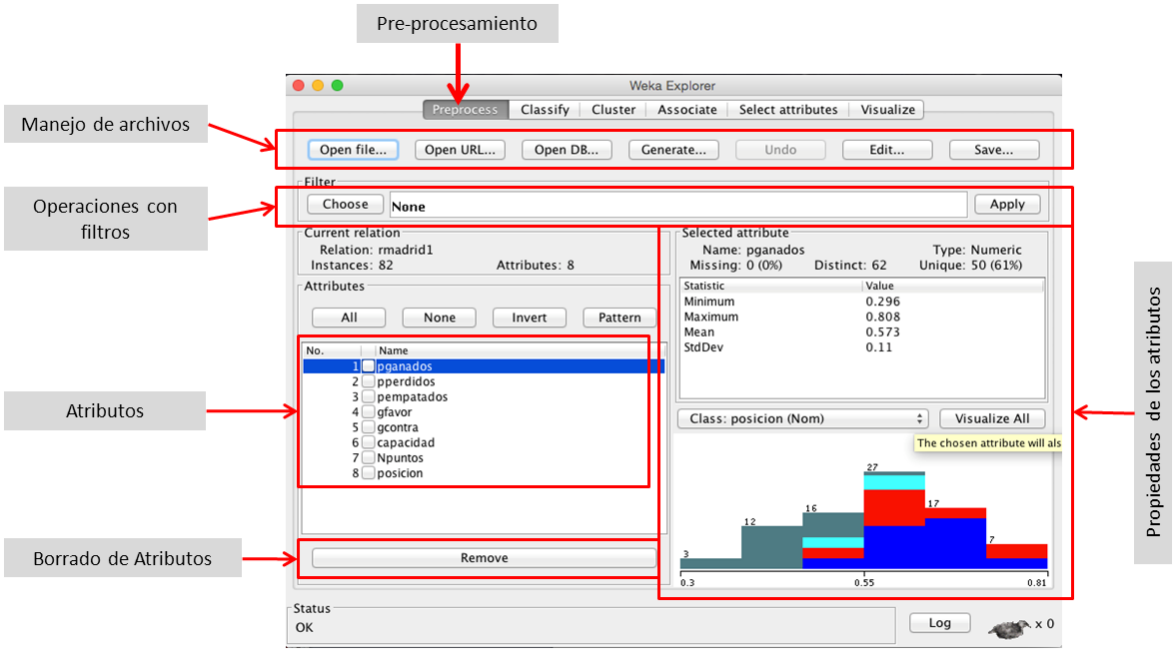


Figura 2.5. Página principal del Explorer.

WEKA permite usar numerosos filtros, por lo que podemos realizar transformaciones de todo tipo sobre nuestros datos. Para poder hacer uso de esta herramienta, se debe seleccionar el botón `choose` (Figura 2.5), donde tendremos acceso a un gran número de opciones, entre las que se encuentran:

- Filtrar atributos.
- Modificar el tipo de atributos (como por ejemplo, discretizar).
- Realizar muestreos sobre los datos.
- Unificar los valores de un atributo.
- Normalizar los atributos numéricos.

En este caso elegiremos, dentro de la carpeta de filtros no supervisados, la opción `atributte` (Figura 2.6), existiendo la posibilidad de utilizar filtros diferentes, entre los destacan:

- `Add`: aporta la posibilidad de añadir un atributo más. Debemos proporcionar el nombre de nuestro atributo, qué posición va a ocupar y los posibles valores separados entre comas. Si estos valores no se especifican el programa supondrá que este nuevo atributo será numérico.

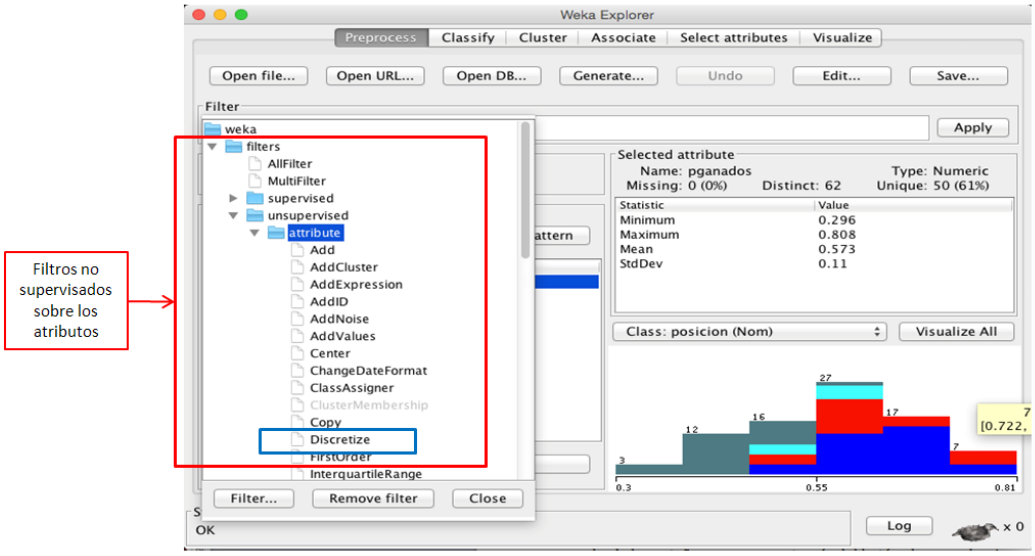


Figura 2.6. Preprocesamientos con filtros.

- `AddExpression`: es uno de los filtros más útiles ya que podemos añadir al final de un atributo el valor de una función.
- `Copy`: realiza una copia del conjunto de atributos en los datos. Este filtro sirve para guardar los atributos originales ya que algunos filtros al utilizarlo destruyen los datos originales.
- `Normalize`: normaliza todos los datos de manera que pasen a tomar los valores 0 o 1.

Como ejemplo de funcionamiento de este apartado, elegiremos uno de ellos, en concreto el filtro de `discretización` que son muy útiles cuando se trabaja con atributos numéricos, que es nuestro caso. El objetivo será convertir los atributos numéricos en simbólicos dividiendo todo el intervalo donde se encuentra definido el atributo en subintervalos más pequeños que definan "características" diferentes. Por ejemplo, la capacidad del estadio en tres categorías, pequeña (hasta 51667 espectadores), mediana (de 51668 hasta 88333), y grande (de 88334 en adelante).

Para lograr este objetivo, seleccionamos el filtro `discretize` (Figura 2.7) y pulsando sobre esta palabra con el botón izquierdo del ratón aparecerá la ventana de propiedades, y modificamos únicamente el apartado correspondiente de `bins` escribiendo un 3. De esta manera hemos discretizado en 3 categorías de la misma amplitud (Figura 2.7).

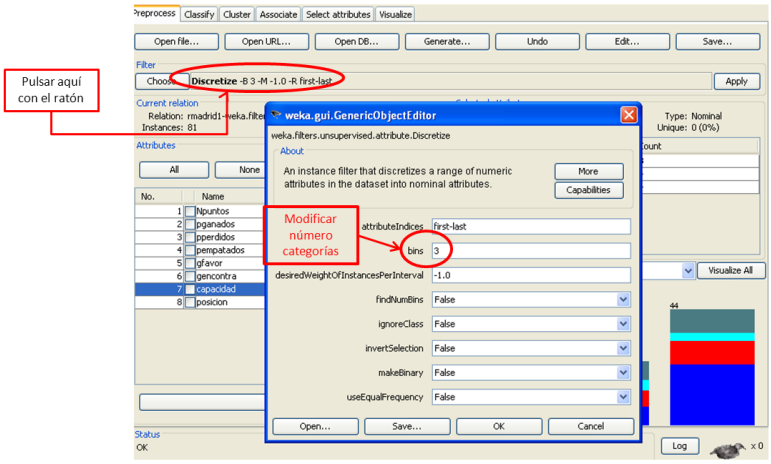


Figura 2.7. Aplicación filtro `Discretize`*.*

2.2.2. Visualización de los datos. El modo de `visualize` nos permite ver gráficamente cómo es la distribución de todos los atributos en gráficas de dos dimensiones, en la que se comparan a través de dicha representación la combinación de los atributos de par en par, permitiéndonos ver correlaciones y asociaciones entre los atributos de forma gráfica.

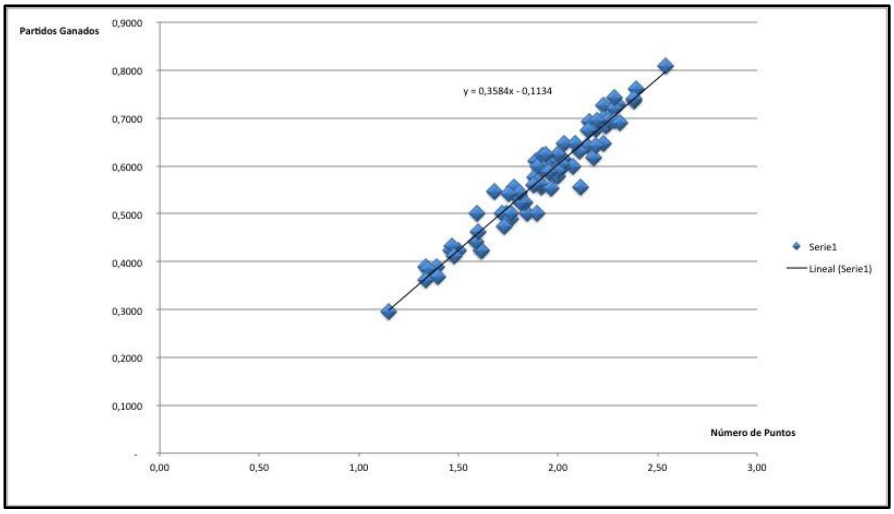


Figura 2.8. Recta de regresión con Excell .

En el caso en el que estemos interesados en encontrar este tipo de correlaciones o asociaciones únicamente para dos atributos cualesquiera, se podrían hacer de forma más sencilla a través de una recta de regresión con el software Excell. Para ello, seleccionamos dos atributos, que en este caso será, por ejemplo, el número de puntos obtenidos (en porcentajes) y el de

partidos ganados. Una vez que elegidos los atributos tendremos un resultado como el que aparece en la Figura 2.8

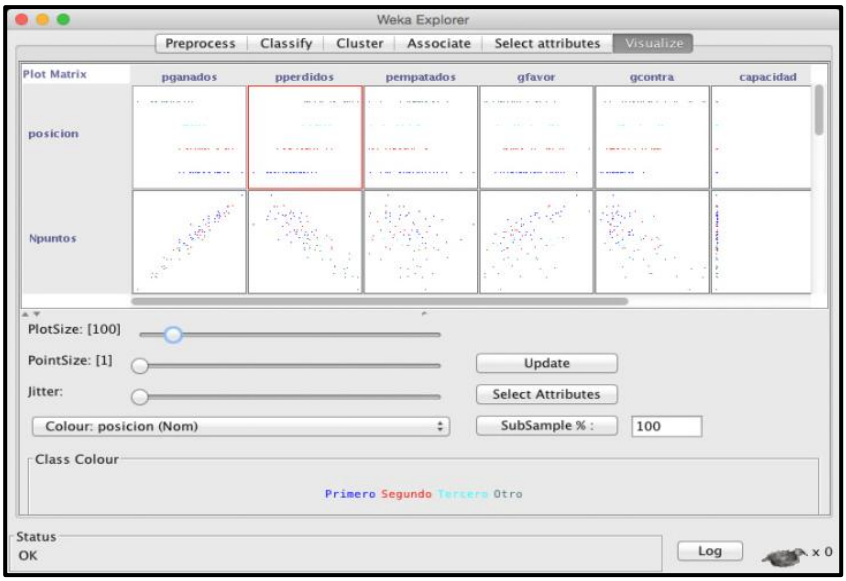


Figura 2.9. Opción visualize de WEKA .

Como puede apreciarse hay una buena correlación entre estos dos atributos existiendo, lógicamente, una relación lineal creciente entre el número de puntos obtenidos y el de partidos ganados.

Ahora bien, esta opción es relativamente sencilla cuando el número de atributos es de dos, en cambio, si el número de atributos es elevado, como es el caso que nos ocupa, esta tarea resulta más complicada. Para ello, cuando disponemos de numerosos atributos, y queremos tener una primera impresión global de las posibles relaciones entre pares de ellos, utilizaremos la opción `Visualize` de WEKA seleccionando la pestaña correspondiente de la pantalla inicial. Al hacerlo podremos observar la ventana que aparece en la figura 2.9.

A través de esta ventana, podemos apreciar todas las comparaciones gráficas posibles entre pares de atributos de nuestra base de datos, con la posibilidad de poder sacar conclusiones sobre algunas de ellas. Por ejemplo, en la figura 2.10, se observa que hay una clara relación entre la posición y los partidos ganados, donde se ve claramente la agrupación en primero, segundo, tercero u otro puesto logrado por el equipo.

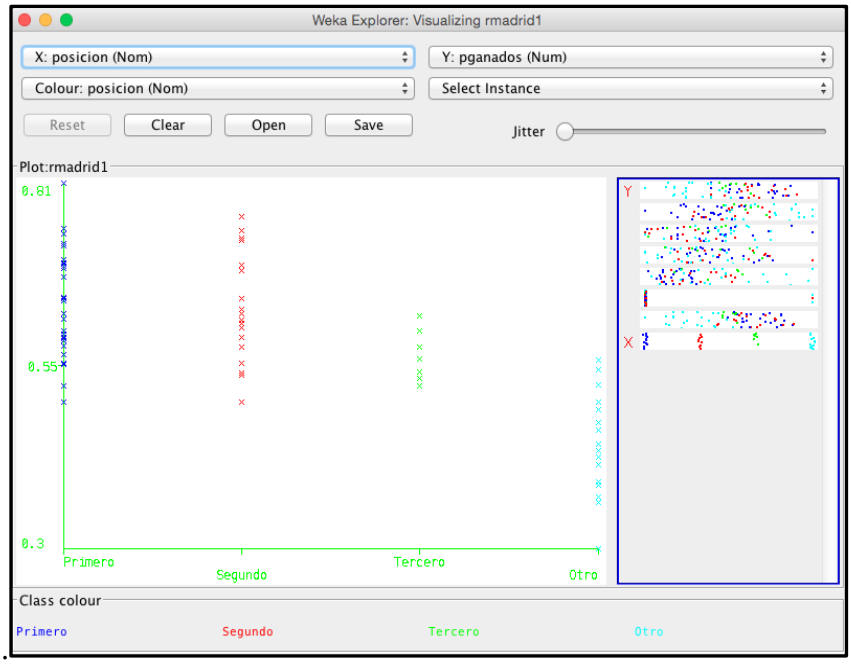


Figura 2.10. Relación visual entre la posición alcanzada (eje X) y los partidos ganados (eje Y).

2.2.3. Clasificación. Cuando estamos interesados en encontrar patrones de comportamientos entre los datos se recurre a la tarea de clasificación, que suele ser la más frecuente entre las realizadas en minería de datos. El objetivo será el de encontrar relaciones entre los atributos que permitan saber cuáles son las posibilidades de que el equipo seleccionado quede en un determinado lugar de la tabla clasificatoria. Esta tarea se lleva a cabo con la pestaña `Classify` (Figura 2.11 izquierda).

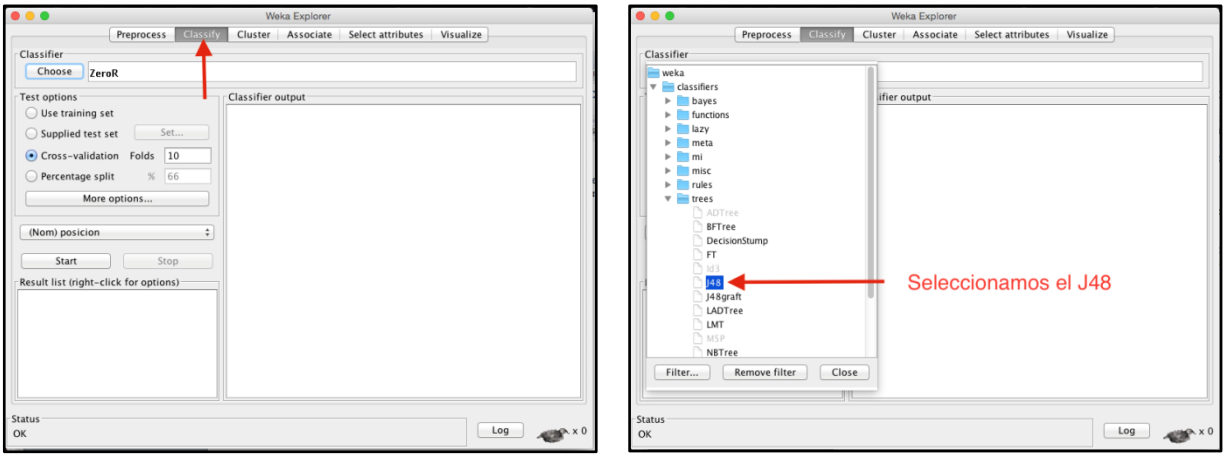


Figura 2.11. `Classify` y árbol decisión `J48`.

Al igual que en el apartado anterior, a través del botón `choose` se puede elegir el método de clasificación que queremos utilizar, entre los que se encuentran (Figura 2.11 derecha):

- **Bayes**: son métodos basados en el aprendizaje de Bayes, que son aquellos que intentan encontrar de entre todas las hipótesis la más probable, a partir de un conjunto de entrenamiento. El algoritmo más utilizado en este apartado es el de `NaiveBayes`.
- **Funciones**: se corresponden con los métodos que están basados en modelos matemáticos, como por ejemplo: las redes neuronales, o los diferentes tipos de regresiones.
- **Lazy**: en este tipo de algoritmos, cada una de las instancias se compara con el resto del conjunto de datos, definiéndose una "medida de distancia", son métodos donde el objetivo es "encontrar al vecino más cercano".
- **Meta**: son los métodos que se obtienen al combinar distintos tipos de aprendizaje.
- **Trees**: métodos expresados a través de árboles de decisión. En este caso se construye un árbol desde la raíz hasta las hojas, de tal manera que las ramas se dividen en función de los valores que toman los atributos. Entre todos ellos, el más popular es el `J48` que es una mejora del árbol inicial `C4.5` diseñado en 1945.
- **Rules**: son algoritmos que se expresan a través de reglas y que tienen la particularidad de ser autoaprendizajes.

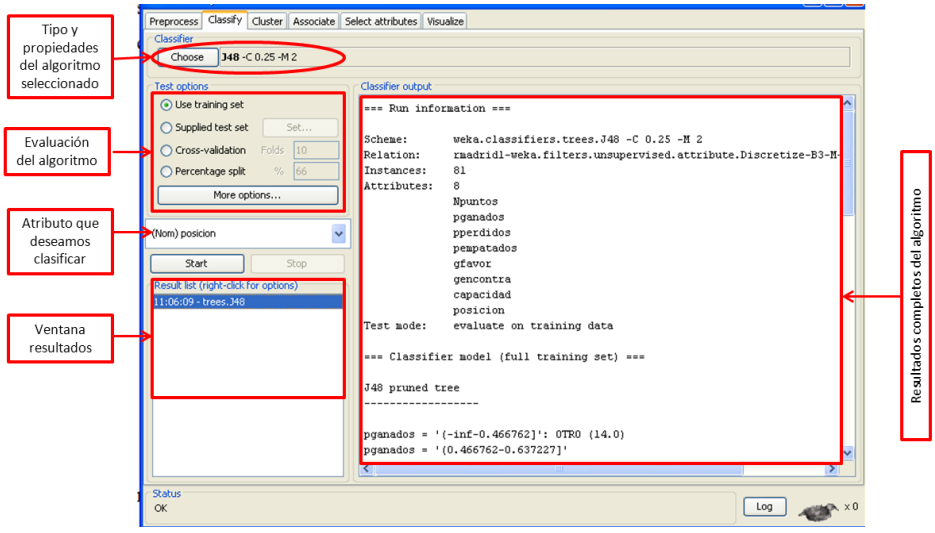


Figura 2.12. Ventana del árbol decisión `J48`.

Además de elegir el tipo de método a usar, en esta ventana (Figura 2.11 izquierda) también existe la posibilidad de elegir el tipo de validación del modelo, que puede ser:

- **Use training set**: con esta opción el programa utilizará el método elegido con todos los datos disponibles y luego realizará una evaluación sobre los mismos datos.
- **Supplied test set**: podemos realizar una evaluación sobre un conjunto de datos que hemos elegido previamente, que normalmente serán distintos a los datos del aprendizaje.
- **Cross-validation**: la evaluación se realizará mediante una técnica de validación cruzada, cuyo objetivo es asegurarse de que los análisis estadísticos realizados son independientes. De todas las posibilidades, esta opción es la que más tiempo computacional consume. Con `Folds` se puede elegir el número de evaluaciones que deseamos llevar a cabo, dividiendo el conjunto de datos en datos de prueba y datos de entrenamiento
- **Percentage split**: en esta última opción podemos definir un porcentaje con el que aprende el modelo, haciéndose la evaluación con los datos restantes.

Como se ha comentado, el método que más se utiliza es el de árboles de decisión, por lo que realizaremos nuestro ejemplo a través de este algoritmo. Para ello pulsaremos sobre el botón `Choose` y entraremos en la carpeta `Trees` para seleccionar el `J48` (Figura 2.12).

Una vez pulsado el botón `Start`, se ejecutará nuestro árbol de decisión. Si no ha habido problemas, el programa nos debe dar la información solicitada por medio de la ventana `Classifier Output`.

El resultado obtenido es el siguiente:

- En un primer apartado se informa del tipo de algoritmo utilizado, el nombre del archivo, el número de atributos, sus nombres, el número de instancias y el modo del test realizado.

```
Scheme:       WEKA.classifiers.trees.J48 -C 0.25 -M 2
Relation:     rmadrid1
Instances:    81
Attributes:   8
              Npuntos
              pganados
              pperdidos
```

```
                pempatados
                gfavor
                gencontra
                capacidad
                posicion
Test mode:      evaluate on training data
```

- A continuación se facilita el resultado del clasificador. Se ha obtenido un árbol de decisión de tamaño 23 y un número de hojas de 12.

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

J48 pruned tree

Npuntos <= 1.738095: OTRO (19.0)

Npuntos > 1.738095

|   Npuntos <= 1.918919

|   |   capacidad <= 15000: SEGUNDO (4.0)

|   |   capacidad > 15000

|   |   |   Npuntos <= 1.76087: SEGUNDO (2.0)

|   |   |   Npuntos > 1.76087

|   |   |   |   pempatados <= 0.23913: TERCERO (9.0/2.0)

|   |   |   |   pempatados > 0.23913

|   |   |   |   |   Npuntos <= 1.825: OTRO (2.0)

|   |   |   |   |   Npuntos > 1.825: PRIMERO (4.0)

|   Npuntos > 1.918919

|   |   gencontra <= 1.217391

|   |   |   gfavor <= 2.021739

|   |   |   |   pempatados <= 0.1875: SEGUNDO (4.0)

|   |   |   |   pempatados > 0.1875

|   |   |   |   |   capacidad <= 90800

|   |   |   |   |   |   Npuntos <= 2.019231: PRIMERO (4.0/1.0)

|   |   |   |   |   |   Npuntos > 2.019231: SEGUNDO (2.0)

|   |   |   |   |   capacidad > 90800: PRIMERO (3.0)

|   |   |   gfavor > 2.021739: PRIMERO (24.0/3.0)
```

```
|   |   gencontra > 1.217391: SEGUNDO (4.0)
```

Number of Leaves: 12

Size of the tree: 23

Time taken to build model: 0 seconds

- Posteriormente se ofrece un resumen del test realizado, donde lo destacado es **que ha clasificado correctamente 75 de las 81 instancias (es decir un 92.6 %) con unos niveles de error muy bajos.**

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

=== Summary ===

Correctly Classified Instances          75               92.5926 %
Incorrectly Classified Instances         6                7.4074 %
Kappa statistic                          0.8943
Mean absolute error                      0.0622
Root mean squared error                  0.1764
Relative absolute error                 17.6828 %
Root relative squared error             42.1157 %
Total Number of Instances               81
```

- La información global de la precisión obtenida también se ofrece para cada una de las clases del atributo clasificado. Podemos observar, por ejemplo, que la posibilidad de quedar en un puesto posterior al tercero tiene una probabilidad de acierto del 95.5%

```
                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   0.969     0.082      0.886     0.969     0.925      0.968    PRIMERO
                   0.8       0          1         0.8       0.889      0.966    SEGUNDO
                   1         0.027      0.778     1         0.875      0.986    TERCERO
                   0.955     0          1         0.955     0.977      0.997    OTRO
Weighted Avg.      0.926     0.035      0.936     0.926     0.926      0.977
```

- Para finalizar, aparece la matriz de confusión. Se trata de una matriz de orden 4x4, donde en la fila i=1, 2, 3, 4 se encuentran las instancias que son de la clase i, mientras que en la columna j=1, 2, 3, 4 aparecen las instancias predichas por el algoritmo. Las 7 instancias mal clasificadas se corresponden a cuatro cuyo puesto es el segundo y se clasifican como primero, uno que se encuentra en el primer puesto pero que se clasifica como tercero y

una última instancia que realmente se encuentra en un lugar posterior al tercer puesto en la tabla pero que se adjudica al tercer puesto.

```
 a  b  c  d   <-- classified as
31  0  1  0 |  a = PRIMERO
 4 16  0  0 |  b = SEGUNDO
 0  0  7  0 |  c = TERCERO
 0  0  1 21 |  d = OTRO
```

El árbol de decisión expresado de la forma que aparece anteriormente es muy confuso. WEKA ofrece la opción de visualizarlo de manera gráfica. Para ello es necesario pulsar sobre el resultado (`Result list`) con el botón derecho del ratón y tendremos acceso al gráfico de la Figura 2.13.

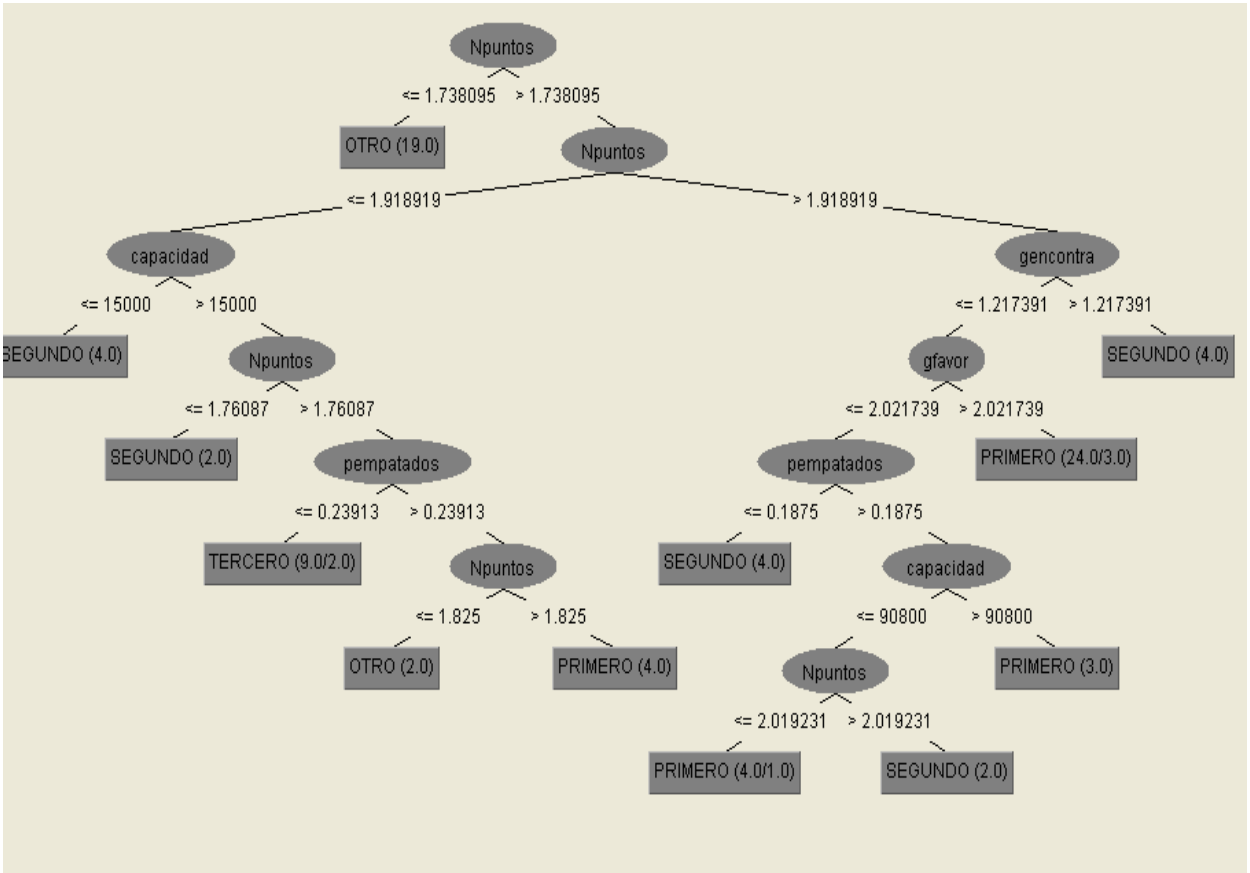


Figura 2.13. Resultado del árbol decisión `J48`.

A la vista del gráfico anterior podemos destacar lo siguiente:

- Si el porcentaje del número de puntos se encuentra por debajo de 1.74, entonces existe una probabilidad del 77% de que el equipo no alcance los tres primeros puestos de la clasificación.
- En caso contrario el porcentaje que decidiría la clasificación sería del 1.92, en cuyo caso los atributos claves serían la capacidad del estadio o los goles en contra.
- Si nos centramos en los goles en contra (puesto que la capacidad del estadio viene impuesta), entonces, si el porcentaje del número de puntos es mayor de 1.92 y el de goles en contra es mayor de 1.22, cada 4 veces de 7 quedaría en el segundo puesto, y si es menor de 1.22, la clasificación dependerá de los goles a favor.
- Bajo las condiciones del apartado anterior, si el porcentaje de los goles a favor es mayor de 2.02, entonces existe una probabilidad del 75% de que el equipo quede líder de la clasificación.

Como hemos podido apreciar, al ser muchos el número de atributos se hace difícil conseguir un árbol que permitan unas reglas más sencillas. Podemos simplificar pero a costa de perder precisión. La pregunta clave sería, **¿cuáles son los atributos que podemos eliminar al ser los que menos influencia ofrecen en la predicción?** El programa dispone de una opción que da respuesta a esta pregunta.

Elegimos la pestaña `Select attibutes` (Figura 2.14) y dejamos los mismos algoritmos de avaluación de atributos y el método de búsqueda que aparecen por defecto, a continuación elegimos como modo de selección de atributos el de validación cruzada y pulsamos el botón `Star`. El programa ofrece como respuesta el porcentaje correspondiente a cada uno de los atributos y observamos que con un valor superior al 90% se encuentran los atributos `Npuntos, pganados y pperdidos`.

Con esta información, volvemos a la pestaña inicial de `Preprocess,` marcamos los atributos que no deseamos (4, 5, 6 y 7) y pulsamos el botón `Remove`.

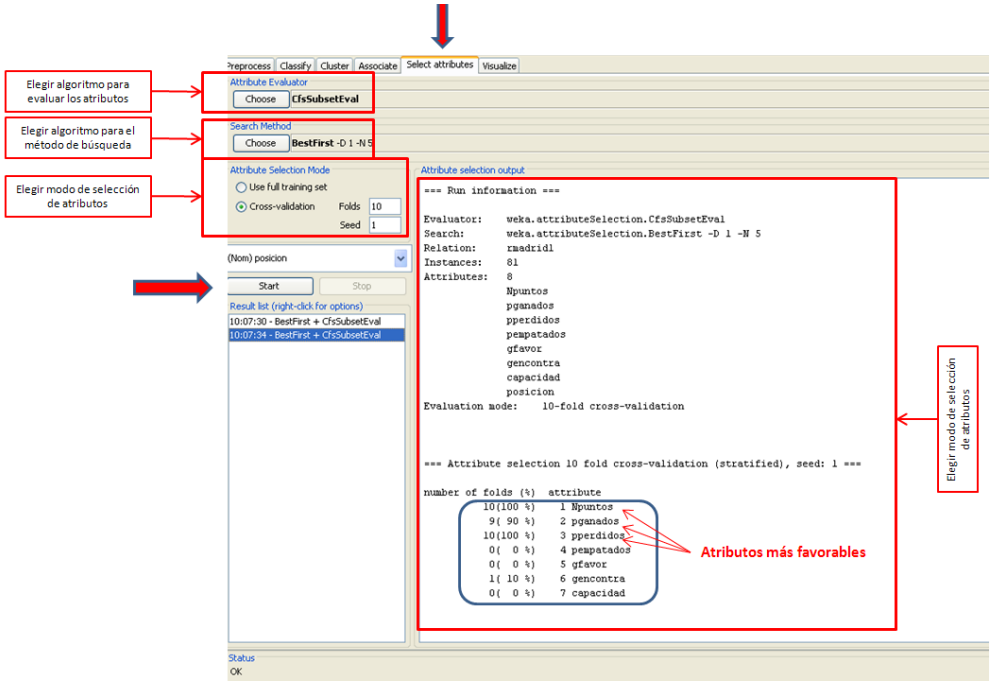


Figura 2.14. Selección de los atributos más importantes .

Finalmente repetimos los pasos ya comentados del clasificador J48 y obtenemos como resultado más destacado:

- El nuevo árbol de decisión tiene un tamaño de 11 y un número de hojas de 6
- El porcentaje de instancias clasificadas correctamente es del 76.5% (que sigue siendo un valor aceptable)
- Los errores cometidos por atributos son:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.306	0.681	1	0.81	0.861	PRIMERO
	0.15	0	1	0.15	0.261	0.77	SEGUNDO
	0.857	0.054	0.6	0.857	0.706	0.964	TERCERO
	0.955	0	1	0.955	0.977	0.997	OTRO
Weighted Avg.	0.765	0.126	0.839	0.765	0.711	0.884	

- La matriz de confusión clasifica 14 segundos como primero, 3 segundos como tercero, 1 tercero como primero y 1 que no se encuentra entre los tres primeros como tercero.
- En este caso, el árbol de decisión es mucho más simple de analizar y se encuentra en la Figura 2.15

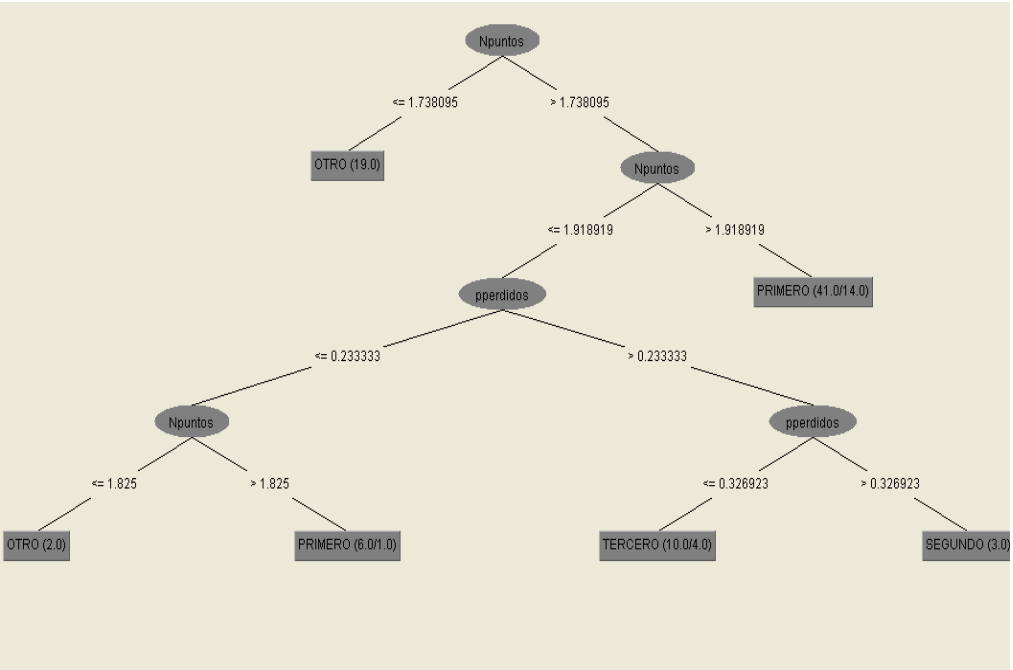


Figura 2.15. Árbol de decisión J48 con sólo tres atributos .

Conclusiones generales:

1. Para quedar primer clasificado es conveniente que el porcentaje de puntos sea superior al 1.92
2. Para alcanzar un puesto entre los tres primeros clasificados el porcentaje de puntos tiene que ser superior a 1.73
3. Si el porcentaje de partidos perdidos es mayor de 0.32 la clasificación más probables es la de segundo puesto y si este porcentaje se encuentra entre 0.23 y 0.32 su posición en la tabla será probablemente el tercer puesto.

2.2.4. Agrupamiento. Las técnicas de `Cluster` permiten realizar agrupamientos de las instancias de la base de datos basándose en las semejanzas y diferencias que existen entre los datos que componen la muestra.

Como ya tenemos cargado nuestro archivo, nos vamos a la pestaña `Cluster`, donde al igual que en las anteriores pestañas volvemos a pinchar (Figura 2.16) en el botón `Choose` y escogeremos un algoritmo, entre los que se encuentra el de `SimpleKMeans`, que es uno de los más utilizados por su sencillez.

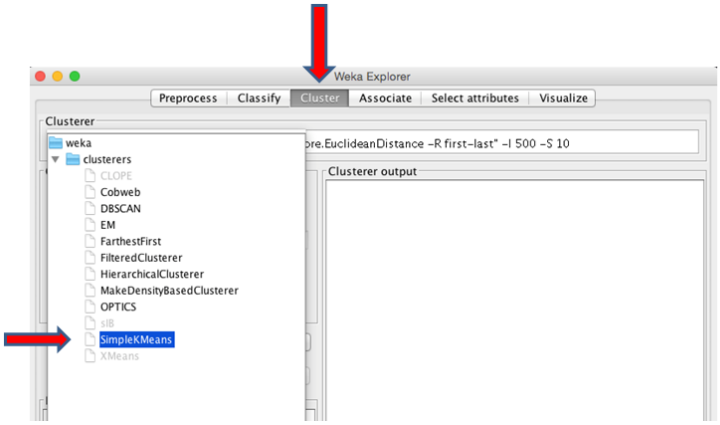


Figura 2.16. Algoritmos de agrupamiento .

El paso siguiente (Figura 2.17) será el de seleccionar los atributos que queremos que entren en nuestro estudio, ya que por ejemplo, la capacidad del estadio no nos aporta ninguna información útil. Para ello pinchamos en `Ignore Atributtes` y seleccionamos capacidad.

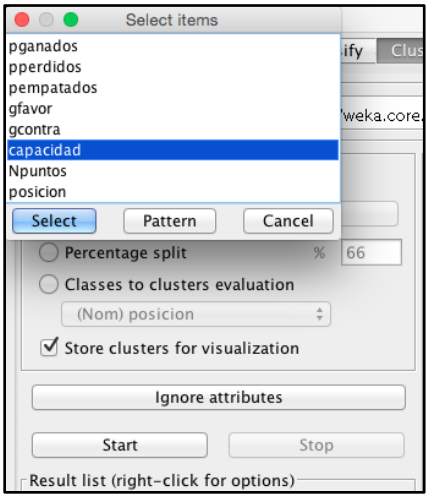


Figura 2.17. Selección de atributos en el algoritmo SimpleKmeans .

Procedemos de manera similar a como lo hicimos con la técnica de clasificación (Figura 2.12). En primer lugar, podemos modificar las propiedades del algoritmo pulsando con el ratón sobre su nombre, por ejemplo, se puede modificar el número K de cluster que deseamos hacer, de esta manera el programa seleccionará de forma aleatoria k instancias que representarán el centro de cada uno de los agrupamientos. A continuación seleccionamos como modo del cluster, `Use training set`, de esta manera usamos la misma muestra como entrenamiento y como

comprobación del resultado obtenido. Finalmente, pulsaremos sobre el botón `start` para ejecutar el algoritmo.

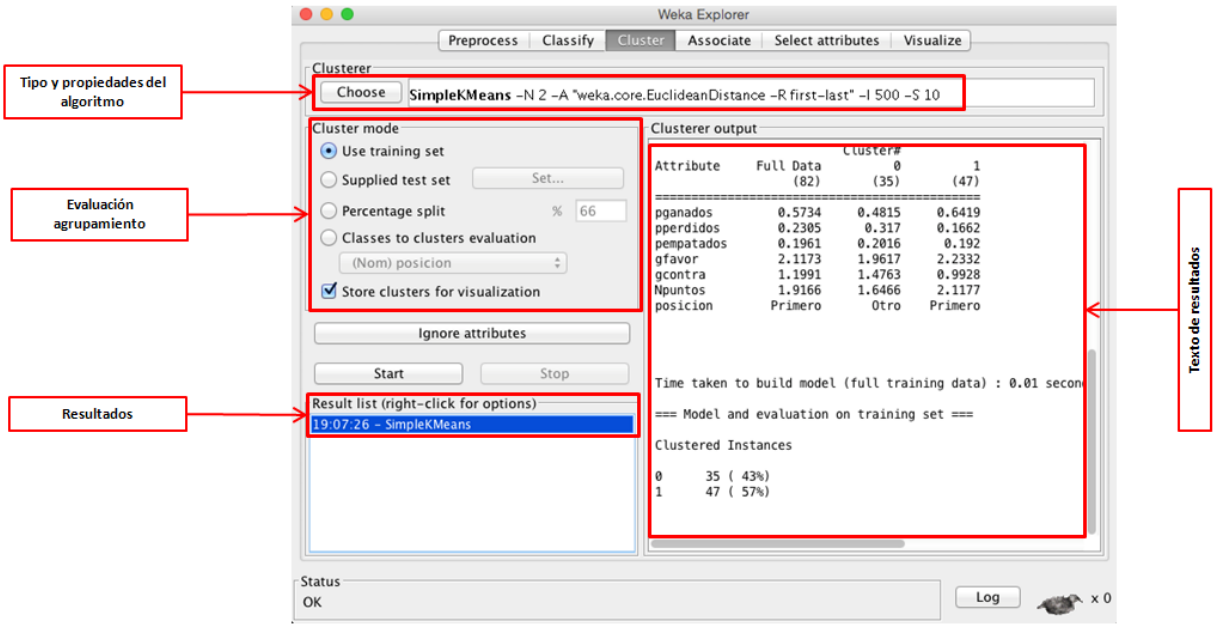


Figura 2.18. Resultados de la ejecución del algoritmo SimpleKmeans .

Como podemos observar, se han obtenido dos agrupamientos con 35 instancias y 47 instancias respectivamente, donde se nos indica cuáles deben ser los porcentajes para que el equipo al final de la liga quede como primer clasificado, o bien no llegue a alcanzar los tres primeros puestos.

Attribute	Full Data (82)	0 (35)	1 (47)
pganados	0.5734	0.4815	0.6419
pperdidos	0.2305	0.317	0.1662
pempatados	0.1961	0.2016	0.192
gfavor	2.1173	1.9617	2.2332
gcontra	1.1991	1.4763	0.9928
Nputos	1.9166	1.6466	2.1177
posicion	Primero	Otro	Primero

Como en el caso anterior, existe la posibilidad de ver estos dos cluster de forma gráfica (Figura 2.19) a través del botón derecho pinchando en `Visualuze Cluster Assignments`. Con esta pantalla podremos generar múltiples gráficas eligiendo cualquier tipo de combinación para cada uno de los ejes.

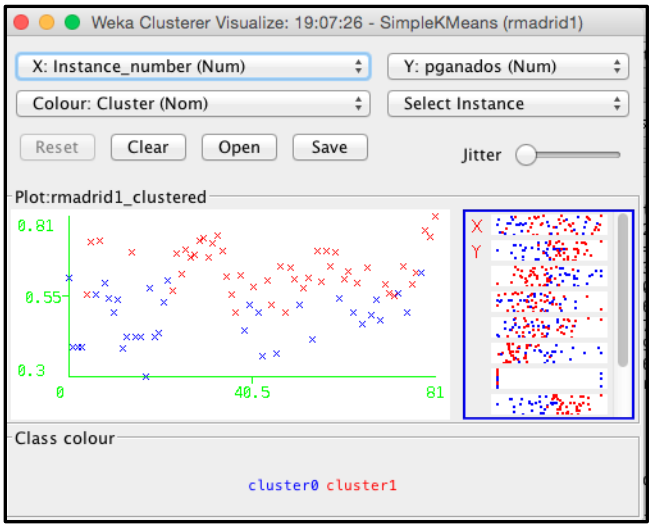


Figura 2.19. Resultados gráficos del algoritmo SimpleKmeans .

2.2.5. Asociación de los datos. Otra de las pestañas, más sencilla de manejar ya que apenas tiene opciones, que podemos usar en WEKA es la de `Associate`. Con esta ventana se aplican los métodos y algoritmos para buscar asociaciones entre los datos. La forma de utilizarla es la siguiente: se selecciona el método, se configura y lo ejecutamos con el botón `Start`. Pero es importante saber que para poder hacer uso de unos de los algoritmos más populares, el de `Apriori`, previamente se tiene que preprocesar los atributos para poderlos discretizar, ya que en caso contrario el programa no te permite seleccionar esta opción.

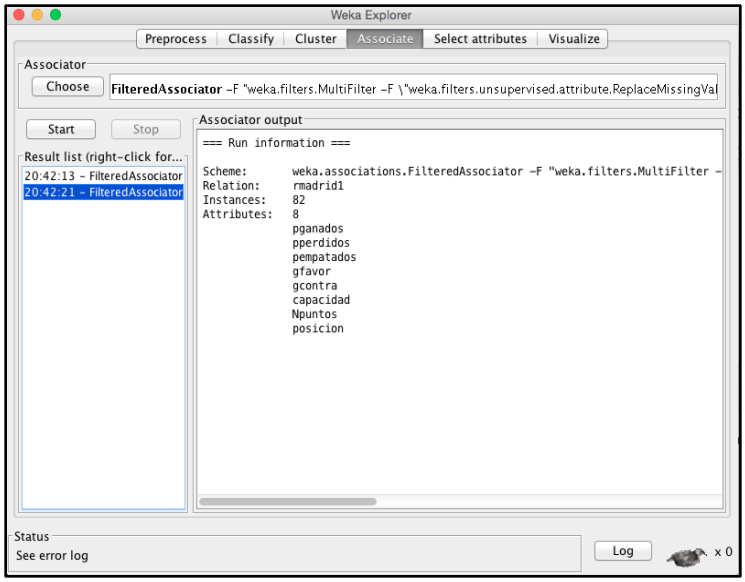
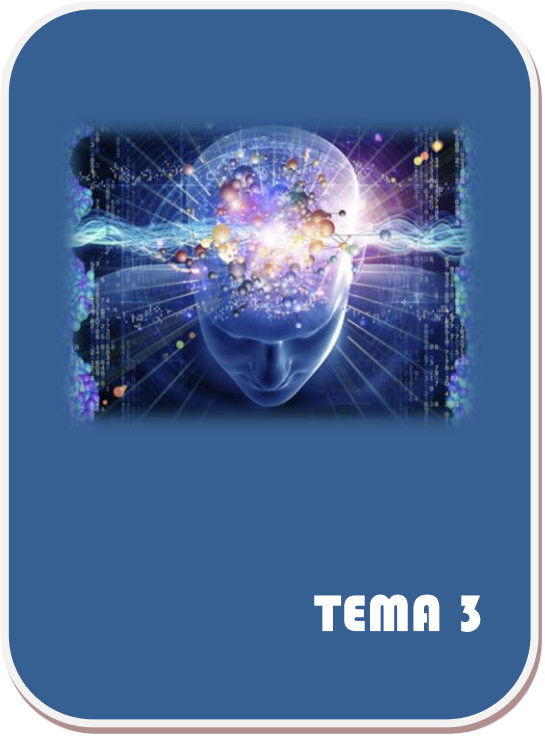


Figura 2.20. Resultados del algoritmo de asociación .



APLICACIÓN DE LA MINERÍA DE DATOS A UN PROBLEMA ECONÓMICO

3.1.- Introducción.

Una vez visto, en el tema anterior, un ejemplo práctico y detallado de cómo funciona el software WEKA, nuestro objetivo en este último tema es el la aplicación de la MD para el caso de un problema relacionado con la economía. Nuestra idea, desde un primer momento, era hacer un estudio relacionado con nuestra tierra, y no hay otra cosa que más destaque sobre ésta que el aceite de oliva. Pero el gran problema con el que nos encontramos es que no existen bases de datos disponibles con la que poder trabajar. Ante este grave inconveniente, pensamos en algún otro producto alternativo, cuya metodología de estudio fuese muy similar, y en esta línea de razonamiento creemos que el vino sería el producto ideal. No obstante, creemos que sería muy

interesante que en el futuro pudiésemos extender, a través de un Proyecto de Investigación, parte de este trabajo al campo de la calidad de los aceites de oliva.

Por lo tanto, el objetivo del tema será aplicar la MD, utilizando WEKA como software, para encontrar patrones y modelos entre la calidad de los vinos de una comarca, y distintos atributos (sulfatos, grado de alcohol, ph, densidad, etc) que, creemos, condicionan la característica del vino.

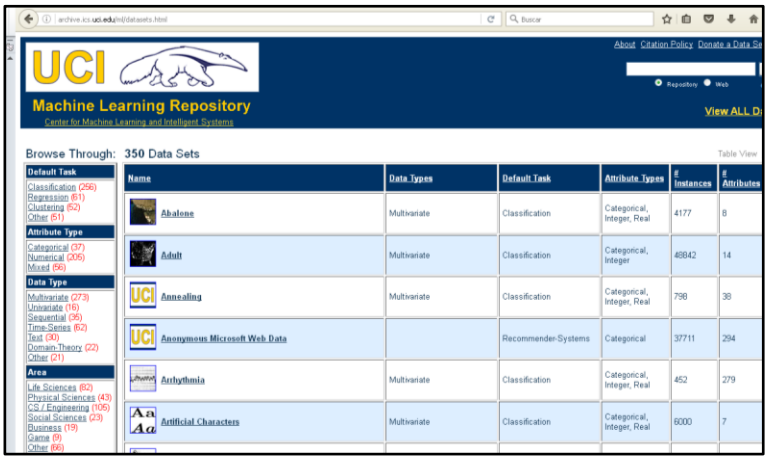


Figura 3.1. http://archive.ics.uci.edu/ml/datasets.html,

Haciendo una búsqueda exhaustiva en la red, hemos encontrado, figura 3.1, una base de datos para diferentes áreas de conocimiento, entre ellas 19 en los negocios.

Figura 3.2. Base datos de calidad del vino

Como hemos comentado, hemos elegido la base de datos correspondiente a la calidad del vino portugués "Vinho Verde" en su variante blanco, realizado en 2009 por *Paulo Cortez (University of Minho), A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of*

de Vinho Verde Region (CVRVV), Porto, Portugal. Con estos datos los autores publicaron un trabajo (Cortez, 2009), donde su objetivo era el de construir modelos para predecir los resultados obtenidos en una cata de vino, utilizando para ello tres técnicas diferentes de regresión.

La base de datos original consta de 1592 instancias y 12 atributos numéricos:

1. **Acidez fija.**
2. **Acidez volátil.**
3. **Ácido cítrico.**
4. **Azúcar residual**.
5. **Cloruros**: mide el grado de oxidación del vino.
6. **Dióxido de azufre libre**.
7. **Dióxido de azufre total**.
8. **Densidad**: se relaciona con la cantidad de alcohol, cuanto mayor sea el grado de alcohol menos será su densidad.
9. **PH**: este valor nos indica el grado de acidez o basicidad del vino.
10. **Sulfatos**: es la cantidad de conservante que posee el vino.
11. **Alcohol**: se le relaciona con la densidad, cuanto mayor sea la densidad menor será su grado de alcohol.
12. **Calidad**: es el conjunto de propiedades del vino que hacen que se pueda caracterizar y valorar con respecto a otros vinos. Los catadores otorgan una calificación numérica de 0 a 10 puntos.

3.2.- Análisis.

3.2.1.- Construcción del archivo arff. Para empezar nuestro estudio, antes de nada, tendremos que realizar algunas simplificaciones a la base de datos original. La primera de ellas, y la más importante, se han clasificado la calidad del vino tinto en tres tipos cualitativos: malos (calificados de 0 a 4), regulares (calificados de 5 a 7) y buenos (los calificados de 8 en adelante). El atributo original con 10 valores diferentes daría lugar a modelos tan complicados de entender que serían, en la práctica, poco eficaces. La segunda simplificación tiene que ver con el número de instancias, ya que la base de datos inicial "no es balanceada" es decir, el número de instancias correspondientes a cada uno de los valores correspondientes al atributo calidad es muy diferente. Para que la base de datos "sea balanceada" se ha elegido aleatoriamente, para cada tipo de vino,

un número parecido de instancias. Con estas simplificaciones, nuestra base de datos, figura 3.3, dispone de 639 instancias y los 12 atributos anteriormente comentados (Anexo II).

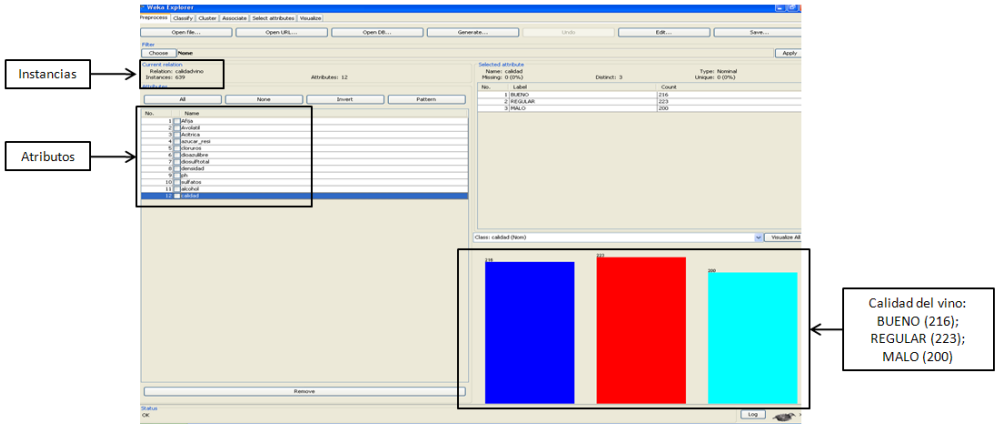


Figura 3.3. Base datos de trabajo.

3.2.2.- Preprocesamiento de los datos

Como ya hemos visto en el tema 2, el preprocesamiento de los datos se realiza desde la pantalla principal que aparece al abrir el archivo con WEKA (figuras 3.3 y 3.4). Con ella podemos ver las propiedades de los atributos, el número de instancias, número de atributos, etc.

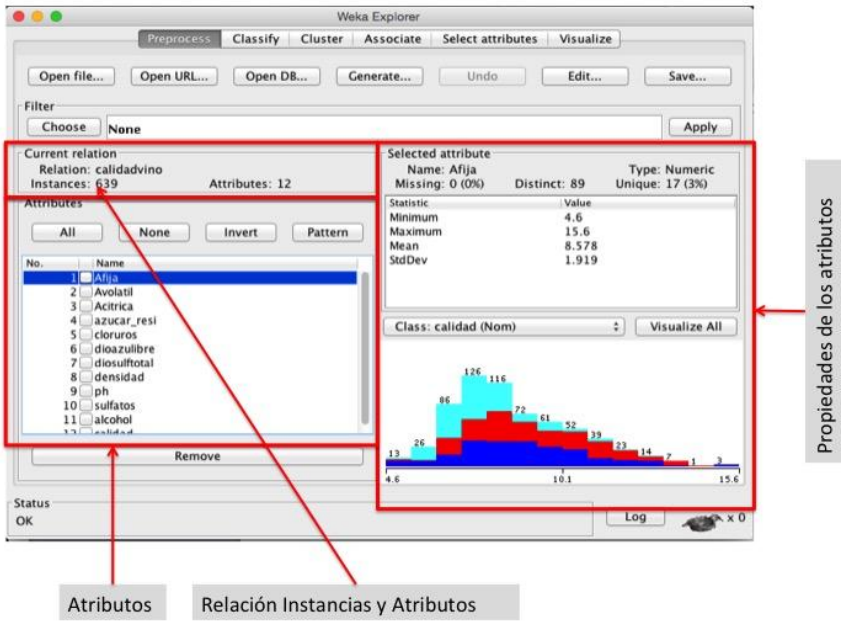


Figura 3.4. Preprocesamiento de los datos.

Si seleccionamos uno de los atributos, podemos observar en la parte derecha de la pantalla (propiedades de los atributos) una parte estadística y otra gráfica. En la primera de ellas aparecen los valores máximos y mínimos que alcanzan cada uno de los atributos, así como su media y su desviación típica, mientras que en la zona gráfica, se encuentran, con colores diferentes, la representación de cada atributo (en nuestro caso seleccionamos la calidad) y su desglose en cada uno de los tipos: bueno, regular y malo (figura 3.3).

3.2.3.- Visualización. A partir de esta pestaña podemos sacar unas primeras conclusiones sobre la relación de los distintos atributos. Como ya hemos explicado, esta pestaña compara gráficamente pares de atributos (figura 3.5), teniendo de esta manera una impresión general de posibles correlaciones, aunque estos patrones y conclusiones preliminares deben ser confirmados con un análisis más detallado. También se ofrece la posibilidad de detectar cuáles son los atributos más importantes para establecer una posterior clasificación.

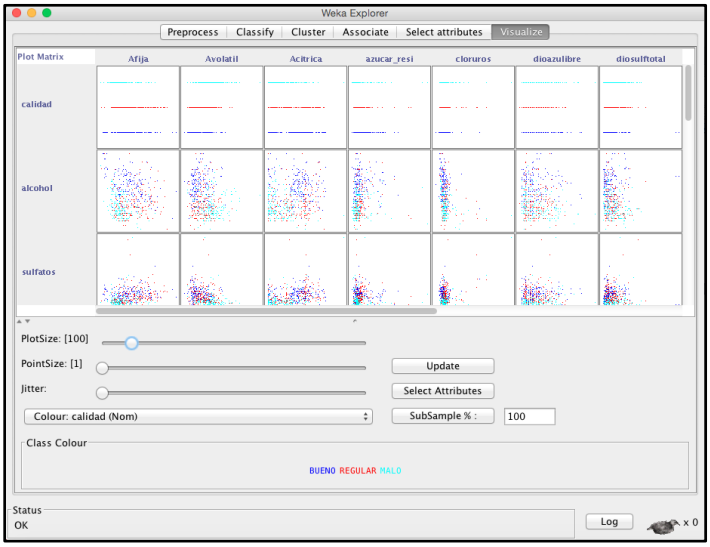


Figura 3.5. Visualización de los datos.

Una vez en la pestaña de visualización, empezaremos con la primera fila comparando la calidad con cada uno de los demás atributos. Por ejemplo, si comparamos la calidad con el grado de alcohol, puede verse en la figura 3.6 izquierda en el eje de abscisas la categoría del vino y en el de ordenadas su graduación alcohólica. De manera general, la graduación se encuentra entre los valores 8.4 y 14, notándose en los vinos considerados como buenos (en azul) que esta

graduación se encuentra ligeramente por encima de los catalogados como regulares o malos (rojos y verdes). En consecuencia, para que un vino se considere como bueno su graduación debe tener como media en alcohol un valor de 11.2

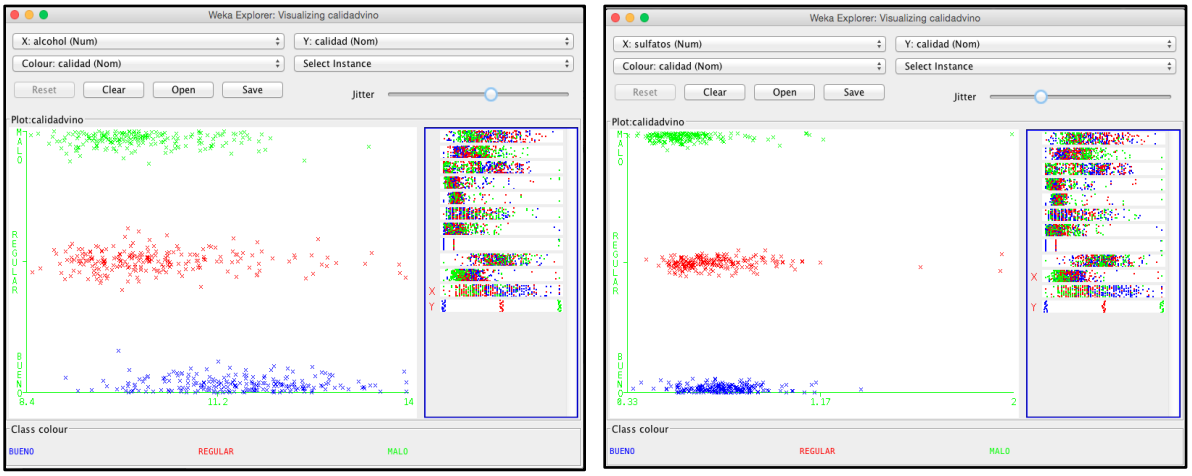


Figura 3.6. Calidad del vino frente al alcohol (izquierda) y frente sulfatos (derecha).

Otro de los atributos que destacan en esta pestaña de visualización es el análisis de la calidad frente a la cantidad de sulfatos. Los valores de los sulfatos oscilan entre 0,33 y 1,17 por lo que al igual que con el grado de alcohol, se puede ver una ligera tendencia entre bueno, regular y malo. Se observa, figura 3.6 derecha, que para un vino bueno los valores de sulfatos son mayores que para un vino considerado como regular o malo.

Por último, el visor gráfico ofrece la posibilidad de configurar el tamaño de los puntos (Jitter), limpiar (Clear), abrir (Open), y grabar (Save) los gráficos, así como seleccionar nuevos atributos en cada uno de los ejes coordenados.

3.2.4.- Algoritmos de clasificación de los vinos. Entre todos los procedimientos de análisis de datos, el de clasificación es el más refinado y utilizado. Recordemos que el objetivo que pretendemos al usar estos algoritmos es el de "predecir" los datos de entrada, en nuestro caso la categoría del vino, teniendo en cuenta el resto de los atributos.

De entre todos los algoritmos de clasificación el más elemental es `OneR`, y debemos ejecutarlo en primer lugar con todas sus opciones por defecto para saber cuáles son los niveles de partidas que debemos superar con otros clasificadores. En validad cruzada con 10 pliegues (folds) el resultado es:

Correctly Classified Instances	367	**57.4335 %**
Incorrectly Classified Instances	272	42.5665 %
Kappa statistic	0.3591	
Mean absolute error	0.2838	
Root mean squared error	0.5327	
Relative absolute error	63.9136 %	
Root relative squared error	113.0599 %	
Total Number of Instances	639	

Se han clasificado correctamente 367 de los 639 instancias (un 57.43%) con unos niveles de error elevados. Si analizamos por cada una de las clases:

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
BUENO	0.495	0.215	0.54	0.495	0.517	0.64
REGULAR	0.735	0.248	0.614	0.735	0.669	0.744
MALO	0.48	0.178	0.552	0.48	0.513	0.651
Weighted Avg.	0.574	0.215	0.57	0.574	0.569	0.68

La clase que mejor clasifica son los vinos considerados regulares y la peor clasificada los vinos malos con un 48% de tasa de aciertos. Estos resultados quedan más claros por medio de la matriz de confusión.

```
   a   b   c   <-- classified as
 107  55  54 |   a = BUENO
  35 164  24 |   b = REGULAR
  56  48  96 |   c = MALO
```

Para mejorar estos resultados hacemos uso de otro clasificador muy popular como es el J48 cambiando dentro de las opciones del test a `Use training set,` (Figura 3.7)

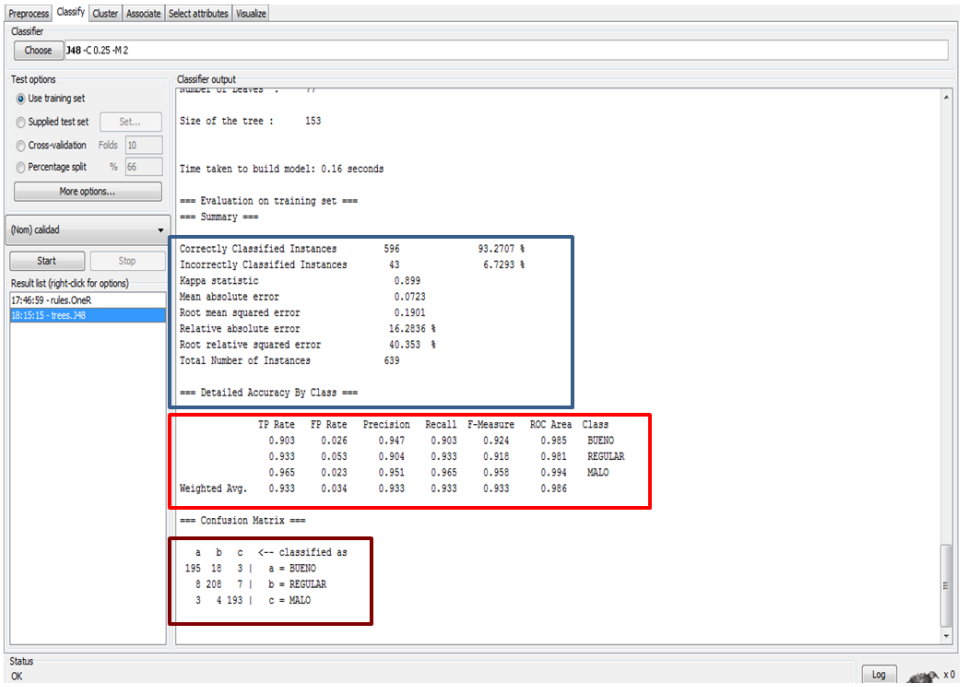


Figura 3.7. Primeros resultados del algoritmo de clasificación J48.

De las 639 instancias que tenemos, WEKA nos ha clasificado correctamente 596 (93.2%) mientras que las restantes 43 (6.8%) se han considerado incorrectas. En concreto, el algoritmo considera:

- Ha clasificado de los 216 vinos que se consideraban de buena calidad 195 buenos, 18 regulares y 3 malos.
- Dentro de los 223 que eran de regular calidad, 8 los ha clasificado como buenos, 208 como regulares y 7 como mala calidad.
- De los 200 que eran de mala calidad, 3 han sido considerados buenos, 4 se han clasificado como regular y 193 malos.

Podemos observar en el rectángulo azul de la figura 3.7, una serie de características, relacionados con la precisión, que son importantes de aclarar algunas de ellas:

1. **El Estadístico de Kappa**: se trata de un índice que permite comparar el nivel de acierto, o ver si el nivel de acercamiento se ha debido al azar. El valor de este parámetro de encuentra entre -1 y 1, siendo el 1 un acercamiento perfecto; y el -1 significa el total desacuerdo. En general, se considera como límite aceptable un 70% o superior, pero los

consideramos muy buenos se encuentran en valores cercanos al 0.9. En nuestro caso el valor obtenido es de 0.899

2. **Mean Absolute Error**: el error absoluto medio es la cantidad que se usa para medir las diferencias entre los cálculos previstos y los observados. El algoritmo ha conseguido un buen valor muy próximo al cero, en concreto 0.0723.
3. **Root Mean Squared Error**: es un número que permite medir la magnitud media del error, es decir, la diferencia existente entre lo que pronosticamos y sus valores observados. El Root Mean Squared Error es siempre mayor o igual al Mean Absolute Error. El valor aportado por el algoritmo J48 es de 0.1901

Al considerar demasiadas instancias, el árbol que hemos obtenido con este algoritmo de clasificación es imposible de interpretar ya que su tamaño 153 con 77 hojas, excede nuestra capacidad de entendimiento.

Por tanto, es necesario hacer uso de algunas técnicas diferentes para que sea más asequible el resultado del clasificador. La primera de ellas es la de disminuir el coeficiente de confianza que se encuentra dentro de las propiedades del algoritmo (figura 3.8) para hacer una primera "poda" del árbol.

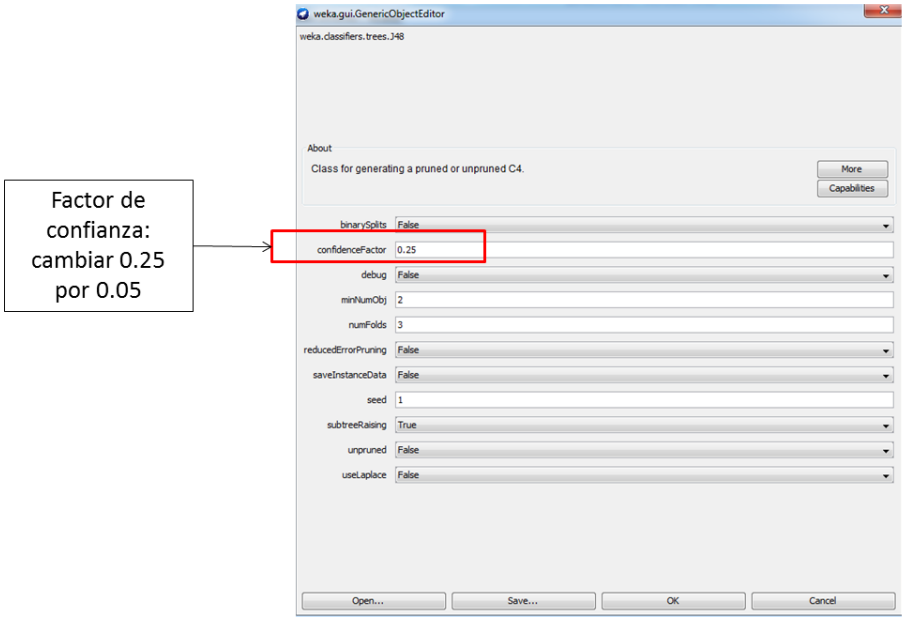


Figura 3.8. Modificación del factor de confianza del algoritmo de clasificación J48.

En la figura 3.9 aparecen los resultados correspondientes al algoritmo *J48* con un nivel de confianza del 5%. En efecto, ahora el tamaño del árbol es de 101 con 51 hojas; siendo su nivel de aciertos del 88% y aceptables mediadas de los errores cometidos, así como la matriz de confusión.

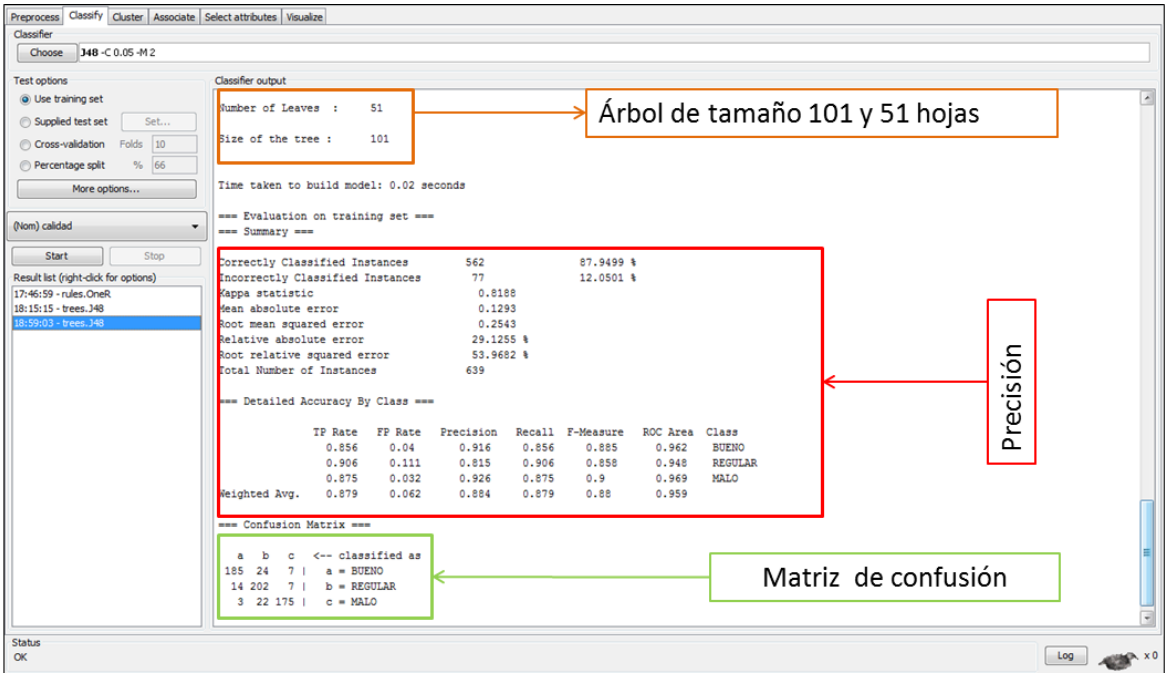
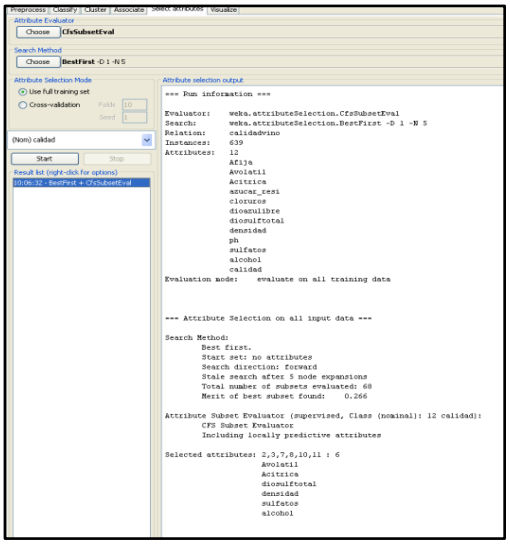


Figura 3.9. Resultados del algoritmo J48 con un factor de confianza de 0.05



Una segunda opción, para poder simplificar el resultado, sería la de elegir aquellos atributos "más significativos" o con "más influencia" en la elaboración del modelo. Para ello elegimos la pestaña `Select attributes` de la pantalla principal, y dejamos por defecto el Evaluador de atributos (`CfsSubsetEval`) y el Método de búsqueda (`BestFirst-D1-N5`). La figura 3.10 muestra el resultado, siendo los atributos 2 (Avolatil), 3 (Acitrica), 7 (diosulftotal), 8 (densidad), 10 (sultatos) y 11 (alcohol) los seleccionados.

Figura 3.10. Selección atributos.

En la pestaña `Preprocess` elegimos con el ratón aquellos atributos que no han sido seleccionados (1, 4, 5, 6 y 9) y los eliminamos con el botón `Remove`. Volvemos a la pestaña del Clasificador y ejecutamos de nuevo el algoritmo *J48* con un factor de confianza del 5%. Ahora la dimensión del árbol es de 95 con hojas 48 y la probabilidad de acierto es de un 85.9% con unos niveles bajos de error (figura 3.11)

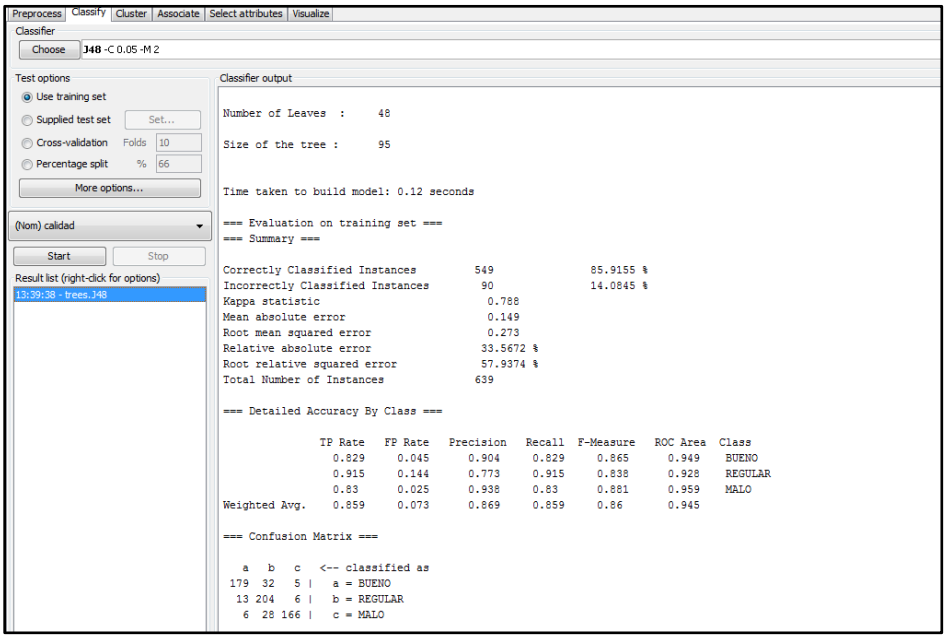


Figura 3.11. Resultados del algoritmo J48 con atributos seleccionados.

Bajo las condiciones anteriores, el algoritmo predice el comportamiento que aparece en el árbol del Anexo III que no permite hacer un análisis simplificado del mismo. No obstante, el modelo proporcionado puede ser programado de tal manera que, a efectos prácticos, los usuarios suministrarían los datos analíticos de entrada y el programa devolvería la clasificación del vino seleccionado.

Con el objetivo inicial presente, y con la idea de ser más operativos, pensamos hacer una nueva simplificación de la base de datos inicial para ver si de esta manera el árbol obtenido podía ser interpretado de una manera más simple. En esta ocasión estudiaremos sólo dos tipos de vinos: buenos y malos. Ahora son 416 instancias, 216 de ellas con un tipo de vino bueno y el resto malo.

Repitiendo todo el proceso de clasificación anterior. Es decir, el algoritmo J48, con un factor de confianza del 5% y todos los atributos; los resultados son bastantes buenos, un 93.5% de aciertos, estadístico Kappa de 0.8701, 0.2436 Root Mean Squared Error, y como matriz de confianza donde 2 de los 200 malos los clasifica como buenos y 15 de los 216 buenos los clasifica como malos.

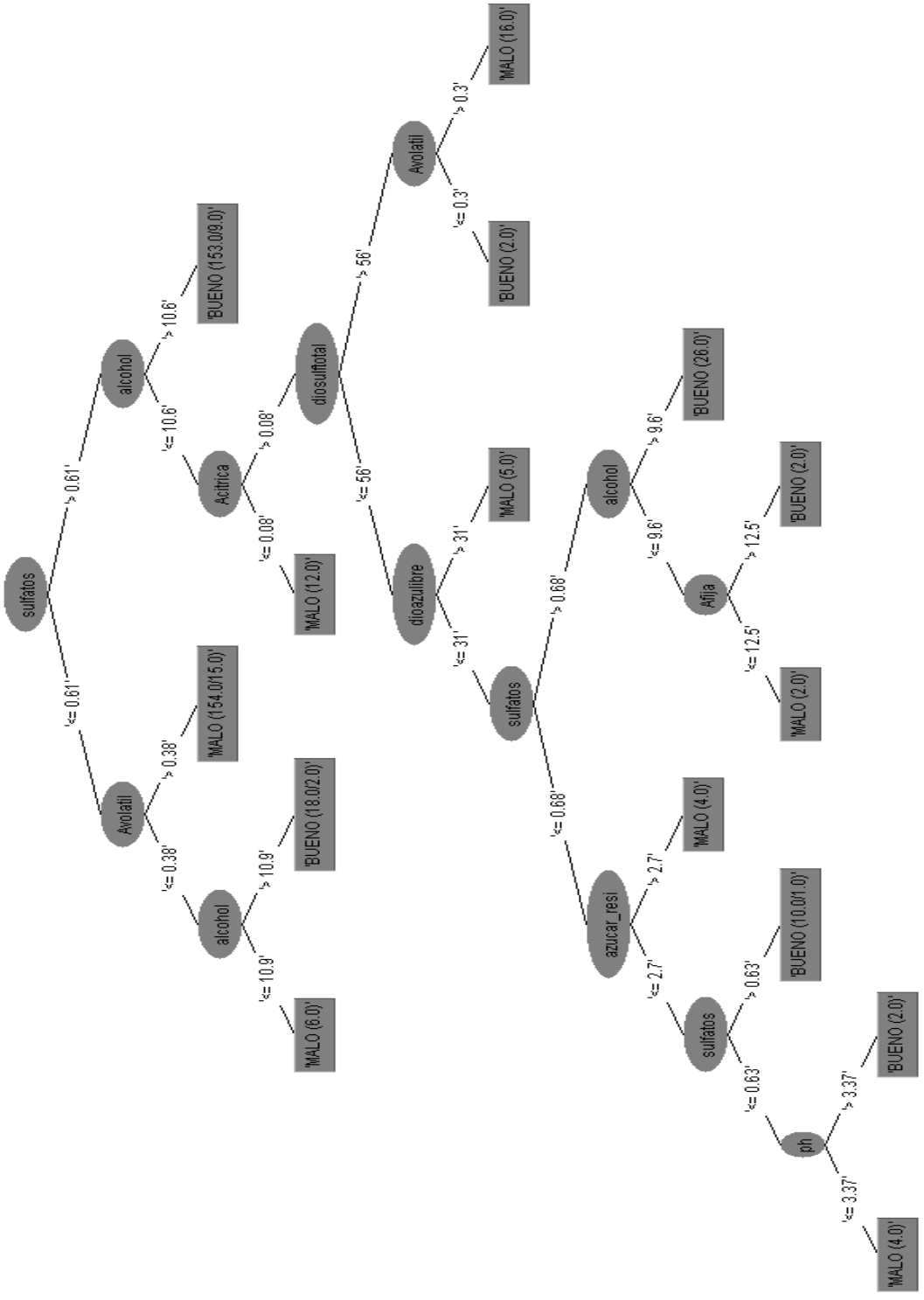


Figura 3.12. Resultados del algoritmo J48 para vinos buenos y malos.

Las conclusiones más importantes que se deducen a la vista del gráfico anterior es que:

1. Si el nivel de sulfatos es menor o igual que 0.61 y el de ácido volátil mayor de 0.38, entonces es "bastante probable" que estemos antes un mal vino.
2. Si el nivel de sulfatos es mayor de 0.61 y el de alcohol se encuentra por encima de 10.6, entonces el vino, probablemente, sea bueno.
3. Para el resto de los casos la casuística es mucho más particular y los resultados analíticos deben compararse con los del árbol de clasificación.

3.2.5.- Agrupamiento (Cluster) El uso de esta técnica nos permite construir grupos de un conjunto de instancias que tienen características similares, por lo que la forma de hacerlo es comparando los distintos valores de los atributos de aquellas instancias que WEKA ha seleccionado. Como comentamos en el tema 2, el programa dispone de numerosas opciones para conseguir este agrupamiento pero el método más utilizado es el denominado `K-Medias` (`SimpleKmeans`) cuyo resultado, una vez ejecutado con los tres tipos de vinos, aparece en la siguiente figura.

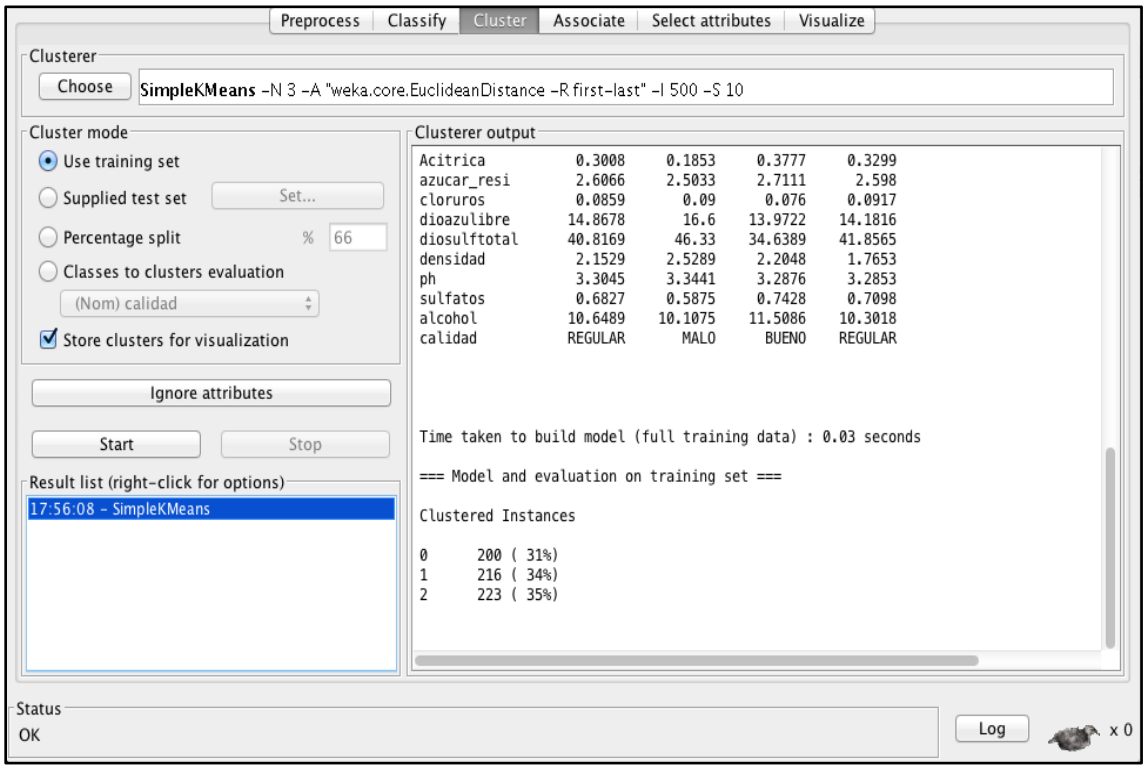


Figura 3.13. Resultados del algoritmo de agrupamiento SimpleKmeans.

Como puede apreciarse en el resultado, hemos configurado, dentro de las opciones del algoritmo, la construcción de tres cluster. Lo interesante del resultado es que cada uno de estos agrupamientos se corresponde con cada uno de los tipos de vinos (Tabla 3.1).

Atributos	Bueno	Malo	Regular
Afija	8,8634	7,5935	9,1857
Avolatil	0,4052	0,6333	0,4850
Acitrica	0,3777	0,1853	0,3299
azucar_resi	2,7111	2,5033	2,5980
cloruros	0,0760	0,0900	0,0917
dioazulibre	13,9722	16,6000	14,1816
diosulftotal	34,6389	46,3300	41,8565
densidad	2,2048	2,5289	1,7653
ph	3,2876	3,3441	3,2853
sulfatos	0,7428	0,5875	0,7098
alcohol	11,5086	10,1075	10,3018

Tabla 3.1. Parámetros medios de cada uno de los cluster.

En la figura 3.14 se aprecia cómo debe ser cada uno de los atributos para que pueda considerarse el vino de una calidad determinada. En la misma figura a la derecha se detecta como uno de los atributos determinantes para calidad del vino es su contenido en alcohol

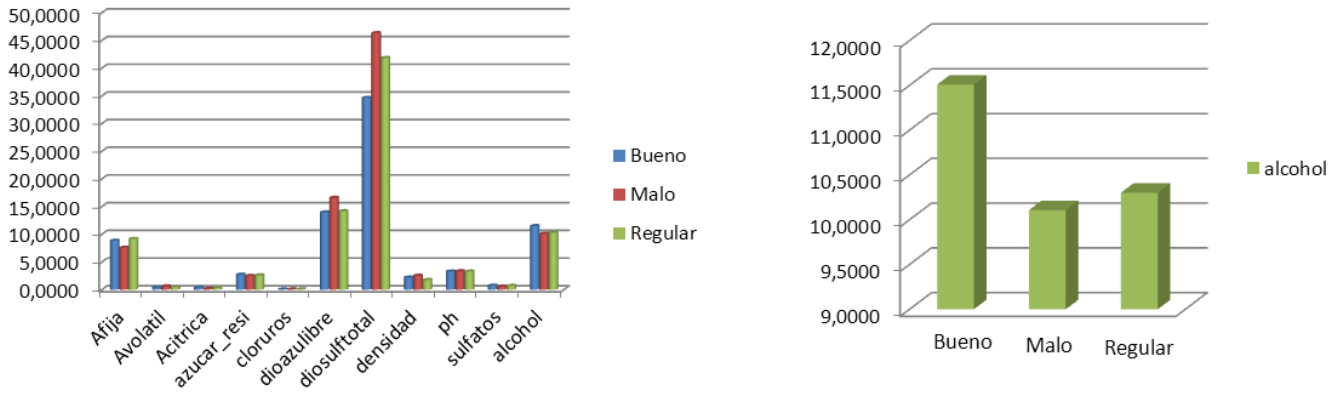


Figura 3.14. Resultados del algoritmo de agrupamiento SimpleKmeans

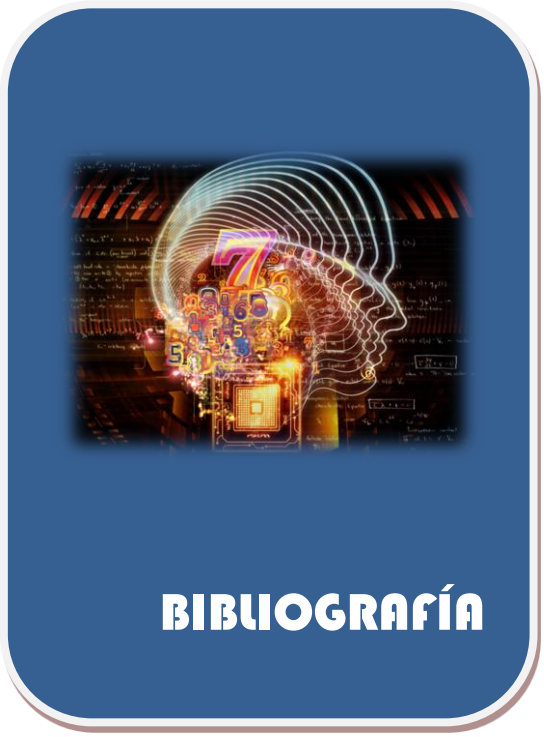
Hoy en día existe una gran cantidad de información almacenadas en potentes bases de datos, pero este enorme volumen excede la capacidad que tiene el ser humano para poder sacarle beneficios directos a los mismos. La Minería de Datos ha ido evolucionando durante el siglo XX, hasta hacerse muy importante en los últimos años.

Vivimos en una época donde diariamente se generan una gran cantidad de datos en campos muy diversos. Poder almacenar y tener acceso a esos datos de forma rápida y clara, junto con la capacidad del ser humano para analizarlos y obtener información valiosa ha permitido el nacimiento de estas técnicas de análisis conocidas como Minería de Datos. No obstante, su potencial reside en la información oculta que podamos extraer de ellos. Al considerar la Minería de Datos como una tecnología emergente, será importante ver como en un futuro no muy lejano se considerará la Minería de Datos como una parte de la empresa ya que con la información que se disponga ayudará a la hora de tomar las decisiones y que éstas sean lo más certeras posibles.

Aunque existen muchas herramientas, el software que hemos utilizado para la implantación de los diversos algoritmos ha sido WEKA, debido a que es un programa de distribución gratuita y además a su facilidad de uso. Por medio de él, hemos tenido la oportunidad de `Preprocesar` los datos usando filtros entre los atributos; `visualizarlos` en la que se comparan la combinación de los atributos de par en par, permitiéndonos ver correlaciones y asociaciones entre los atributos de forma gráfica; `Clasificarlos` con la posibilidad de encontrar patrones de comportamientos entre los datos; `Agruparlos` por medio de las semejanzas y diferencias que existen entre los datos que componen la muestra; y por último la `asociación` de datos.

Considero muy interesante esta introducción a la Minería de Datos ya que me ha posibilitado estudiar y profundizar en un aspecto novedoso del curriculum. El Big Data está de plena actualidad en el mundo empresarial y son muy pocos los profesionales formados en este campo y menos aquellos que completan estos estudios entre las matemáticas, la informática, la inteligencia artificial y la estadística, con conocimientos del mundo de la empresa.

Por otro lado, el trabajo ha dejado la puerta abierta a una futura aplicación de esta metodología, que se ha realizado con el vino, en un campo de tanto interés en nuestra provincia como es el aceite de oliva. Encontrar patrones o modelos matemáticos que nos permitan completar, de una manera menos subjetiva, que la aplicada en la actualidad basada en la cata de los aceites.



- Brodley, C.E.; Lane, T.; Stough, T.M. (1999). *Knowledge discovery and data mining*. *American Scientist*. Vol. 86, 55-65, (1999)
- Cortez, P; Cerdeira, A.; Almeida, F.; Matos, T.; Reis, J. *Modeling wine preferences by data mining from physicochemical properties*. *Decision Support Systems, Elsevier*, 47(4):547-553, 2009
- Hernández-Orallo, J.; Ramírez Quintana, M.J.; Ferri Ramírez, C. *Introducción a la Minería de datos*. Pearson Printice Hall, D.L., (2007).
- Herrera Triguero, F. *Inteligencia computacionales y big data*. Universidad de Jaén, Servicio de Publicaciones, (2014).
- Molina Felix, L.C. *Data mining: torturando los datos hasta que confiesen.* Blog: http://www.uoc.edu/web/esp/art/uoc/molina1102/molina1102.html
- Molina, L.C. *Data mining no processo de extração de conhecimento de bases de dados*. Tesis de master. Sao Carlos (Brasil): Instituto de Ciencias Matemáticas y Computaçión. Universidad de Sao Paulo. (1998).
- Recopilación de aplicaciones de Big Data: http://www.sc.ehu.es/ccwbayes/members/inaki/DM-applications.htm

- Sierra Araujo, Basilio. *Aprendizaje Automático: conceptos básicos y avanzados. Aspectos prácticos utilizando el software WEKA*. Pearson Education. Prentice Hall, (2006).
- Software WEKA: http://www.cs.waikato.ac.nz/ml/WEKA. Curso WEKA Universidad de Waikato: http://www.cs.waikato.ac.nz/ml/WEKA/mooc/dataminingwithWEKA/
- Virseda Benito, F.; Román Carrillo, J. *Minería de Datos y aplicaciones*. Universidad Carlos III. Madrid.
- Written, Ian H.; Frank Eibe; Hall, Mark A. *Data mining: practical machine learning tools and techniques with Java Implementations*. Morgan Kaufmann, (1999).
- Análisis de datos con WEKA: http://isa.umh.es/asignaturas/crss/turorialWEKA.pdf
- Imágenes algunos derechos reservados. Bajo licencia de Creative Commons 3.0 License.
- Base de datos en red: http://archive.ics.uci.edu/ml/datasets.html