Universidad de Granada

Máster Universitario en Estadística Aplicada

Aplicación de técnicas de Minería de Datos a datos obtenidos por el Centro Andaluz de Medio Ambiente (CEAMA)

Trabajo Fin de Máster

Realizado por
Francisco José García González

Tutores:
Silvia González Aguilera
Juan Antonio Maldonado Jurado

Granada, 2013

Contenido

1. Introducción

Hoy día nuestra sociedad genera grandes cantidades de información que unido al aumento de las capacidades de almacenamiento, han hecho que todo tipo de organizaciones puedan disponer de una gran cantidad y variedad de datos relativos a su actividad diaria. Esta información ofrece a la empresa una visión perspectiva (qué se está haciendo y cómo se está haciendo) y prospectiva (cómo puede evolucionar la organización en un futuro a corto-medio plazo) y es por ello por lo que tiene una función vital en el proceso de toma de decisiones.

Sin embargo mucha de la información recogida en las bases de datos no se encuentra bien estructurada resultando difícil de explotar desde el punto de vista estadístico por lo que para su utilización es necesario un proceso de tratamiento y análisis exhaustivo de los datos allí recogidos que llamaremos *minería de datos*.

La *minería de datos* se engloba dentro de un proceso más amplio conocido *como extracción de conocimiento en bases de datos,* si bien algunas veces y debido a que la frontera entre ambos conceptos no es clara, suelen utilizarse como sinónimos.

Imagen 1: Fases del proceso de extracción de conocimiento en bases de datos

En un proceso de *minería de datos* se realizan diferentes tareas:

- Descriptivas: Para identificar patrones que explican o resumen los datos.
- Predictivas: Permiten estimar valores futuros o desconocidos de variables de interés, a partir de otras variables de la Base de Datos.

Para conseguir esos objetivos, el investigador puede utilizar diferentes técnicas como:

- Sistemas de agrupamiento: Consiste en obtener grupos naturales a partir de los datos. También se conoce como segmentación o *clustering*.
- Reglas de asociación: Su objetivo es identificar relaciones no explícitas entre variables categóricas.
- Reglas de asociación secuenciales: Caso particular de las reglas de asociación, en el que se buscan relaciones temporales en los datos.
- Correlaciones: Proporcionan información sobre el grado de similitud entre variables cuantitativas.
- Clasificación: Cada objeto pertenece a una clase, indicada por el valor de un atributo. El objetivo es predecir la clase a que pertenecen nuevos objetos a partir de las restantes variables.
- Regresión: Su objetivo es predecir el valor que toma una variable cuantitativa en nuevos objetos a partir de la información proporcionada por las restantes variables.

2. Software disponible para aplicar técnicas de minería de datos

Actualmente existen varios paquetes y complementos, cada uno con sus propias características, que permiten aplicar diferentes técnicas de minería de datos al conjunto de datos con el que trabajemos. Presentamos en este capítulo algunos de ellos.

2.1. XLMiner

XLMiner es un complemento para Excel, con funcionamiento mediante macros, que permite muchos tipos de análisis tanto para datos de tipo corte transversal, como secuencias temporales.

Entre las principales características de XLMiner se encuentran:

- Manejo de bases de datos, con imputación de datos faltantes.
- Realización de predicciones.
- Modelos ARIMA, Holt winters, Polinomiales.
- Arboles de decisión, análisis clúster.
- Facilidad para la entrega de informes.
- Redes neuronales.

A favor de este programa se puede decir que:

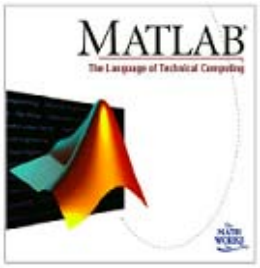
- Posee un buen manual que se encuentra dentro de la sección ayuda, que describe los distintos métodos y parámetros.
- Presenta opciones de configuración y trabajo (interface) amigables para cada método.
- Los formatos de presentación de resultados como gráficos tablas e indicadores de cambios son muy ordenados y tienen buen formato.
- Existen muchos videos tutoriales para los distintos métodos que indican paso a paso qué se necesita hacer.

En contra de este programa tenemos que:

- XLMiner es accesible como herramienta de prueba por un periodo limitado de 30 días.
- Se debe pagar por tener acceso a la versión que no limita el tamaño de la base de datos.
- No posee indicadores de errores claros. Como XLMiner trabaja en base a macros de Excel, al parametrizar alguna operación con datos inadecuados, la operación se interrumpe, siendo imposible de recuperar lo último realizado.

Finalmente el programa puede descargarse desde la web del creador http://www.solver.com/xlminer/, tras rellenar el formulario para la versión prueba que se encuentra al final de la página.

2.2. Matlab



MATLAB (abreviatura de MATrix LABoratory) es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos.

Este programa dispone también de un amplio abanico de programas de apoyo especializado, denominados Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación.

MATLAB también se provee de un lenguaje de programación propio, similar al de otros lenguajes como Fortran o C. A través de este lenguaje, el usuario puede realizar cualquier tipo de regresión disponible o bien crear un proceso de validación cruzada a medida.

En relación a este trabajo destacaremos las siguientes Toolboxes:

- **Statistics Toolbox**: Combina algoritmos estadísticos con interfaces gráficas interactivas.
- **Nnet**: Herramientas para el procesado de redes neuronales. Se subdivide principalmente en:

    - *nnet\nnet - Neural Network Toolbox:*

      La Neural Network Toolbox es un paquete de Matlab que contiene una serie de funciones para crear y trabajar con redes neurolales artificiales. Así pues, proporciona las herramientas para el diseño, la puesta en práctica, la visualización, y la simulación de redes neuronales.

      Las redes neuronales son herramientas de gran alcance en situaciones donde sería difícil o imposible el análisis formal, por ejemplo el reconocimiento de patrones y la identificación y el control no lineales del sistema. La Neuronal Network Toolbox también proporciona una interfaz gráfica que permite diseñar y manejar las redes que el usuario desee. El diseño modular, abierto, y extensible de la Neuronal Network Toolbox simplifica la creación de funciones y de redes. En resumen, como principales características presenta:

- ✓ Interfaz gráfica (GUI) para crear, entrenar, y simular a sus redes neuronales, así como ayuda al usuario de las arquitecturas de redes supervisadas y no supervisadas más comunes.
- ✓ Un sistema sencillo para realizar el entrenamiento y creación de funciones de aprendizaje.
- ✓ Representación modular de la red, permitiendo un número ilimitado de la entrada que fija capas, e interconexiones de la red, así como funciones para mejorar el entrenamiento, funcionamiento y visualización de la misma.

- ○ *nnet\nncontrol - Neural Network Toolbox Control System Functions:*

  Provee un conjunto de funciones para medir y controlar el sistema de redes neuronales construido.

- ○ *nnet\nndemos - Neural Network Demonstrations:*

  Conjunto de muestras de redes neuronales.

Finalmente, se trata de un programa comercial por lo que para su uso se ha de pagar una licencia. Para obtener más información sobre MATLAB se puede acceder a la página oficial de esta herramienta: http://www.mathworks.es/products/matlab/

2.3. IBM SPSS Modeler

Se trata de un producto de la empresa *IBM SPSS* que permite, mediante una interfaz gráfica, aplicar técnicas de minería de datos para descubrir patrones y tendencias en datos estructurados o no estructurados.

IBM SPSS es una empresa reconocida como líder en análisis predictivo. Sus aplicaciones tienen una buena visibilidad y fuerza de ventas en el mercado y es considerada por algunos expertos como uno de los mejores proveedores de software de minería de datos.

Con *IBM SPSS Modeler* se puede visualizar gráficamente el proceso llevado a cabo, así como crear nuevas funciones que se añaden a las ya implementadas. Además se provee de una serie de módulos que permiten realizar un análisis de minería de datos con grandes volúmenes de datos.

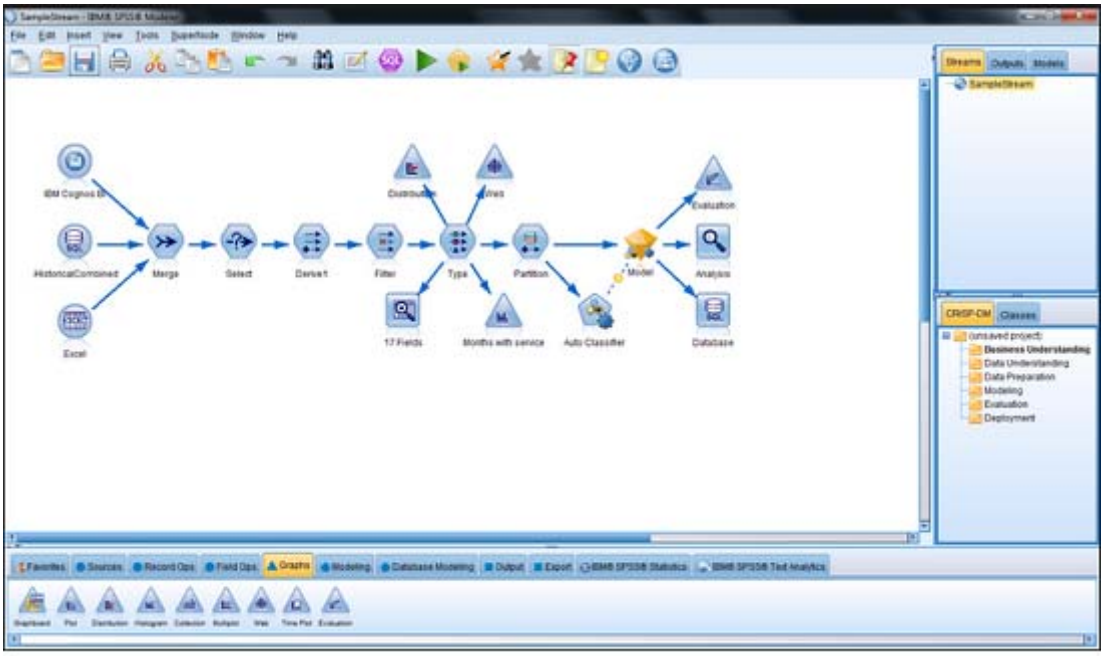


Imagen 2: Diagrama de flujo del proceso con *SPSS Modeler*

En lo referido a técnicas de minería de datos, esta herramienta proporciona diferentes métodos según el proceso que vayamos a realizar; algunas de ellas son:

- *Segmentación*: K-medias, Kohonen, Bietápico, Anomalía.
- *Asociación*: A priori, GRI, CARMA y Análisis de Secuencia.
- *Clasificación*: Factorial, Discriminante, Red Neuronal, C5.0, GLM, Máquinas de Vectores de Soporte, Redes Bayesianas, Modelos de auto aprendizaje, Vecino más próximo, Árboles, Listas de Decisión, Selección de características, etc.
- *Predicción*: Regresión Lineal, Series Temporales, Regresión de Cox, Regresión Logística.
- *Automáticos*: Auto numérico, Auto clasificador, Auto Agrupación, Modelizador ARIMA automático.

Finalmente y al igual que en los programas anteriores se trata de un programa comercial y se ha de pagar una licencia para su uso.

Para obtener mas información sobre *IBM SPSS Modeler* se puede consultar la web del fabricante: http://www-01.ibm.com/software/analytics/spss/products/modeler/

2.4. SAS Enterprise Miner

SAS Enterprise Miner agiliza el proceso de minería de datos para crear modelos predictivos y descriptivos de alta precisión para grandes volúmenes de datos. Ofrece una sencilla interfaz gráfica que integra el conjunto de herramientas necesario para la toma de decisiones.

La solución Enterprise Miner se basa en la metodología *SEMMA* (Sample, Explore, Modify, Model, Assess) desarrollada por *SAS Institute* y puede definirse de la siguiente forma:

- Muestra (Sample): Consiste en identificar los datos.
- Explora (Explore): Su función se traduce en explorar los conjuntos de datos para observar huellas inesperadas, relaciones, patrones, u observaciones inusuales, con nodos para representar los datos, generar una amplia variedad de análisis, identificar las variables importantes, o realizar análisis de asociación.
- Modificar (Modify): Consiste en preparar los datos para el análisis. Los nodos pueden crear variables adicionales o transformar las variables existentes para el análisis mediante la modificación o la transformación de la forma en la que las variables se utilizan en el análisis, filtrar los datos, sustituir los valores perdidos, condensar y contraer los datos en preparación para el modelado de series, o realizar análisis de conglomerados.
- Modelo (Model): Donde se adapta el modelo estadístico. Los nodos predicen la variable objetivo en función de las variables de entrada mediante el uso de cualquier método: mínimos cuadrados o regresión logística, árboles de decisión, redes neuronales, redes dmneural, definido por el usuario, conjunto, vecino más cercano, o el modelado de dos etapas.
- Evaluar (Asses): Donde es posible comparar la exactitud entre los modelos estadísticos, con nodos para comparar el desempeño de los diversos modelos de clasificación mediante la visualización de las estimaciones de probabilidad en competencia de los gráficos de elevación, gráficos ROC y tablas de umbral. Para diseños de modelado predictivo, el rendimiento de cada modelo y los supuestos del modelo pueden ser verificados a partir de las parcelas de predicción y gráficos de diagnóstico.

Entre las principales características de esta herramienta destacan:

- El acceso a los datos, la gestión y la limpieza se integran a la perfección, por lo que es más fácil de preparar los datos para el análisis.
- Alta integración con otras bases de datos debido a la gran experiencia de la empresa para operar con grandes volúmenes de datos.
- Proporciona sólidas herramientas de modificación y selección de los datos lo que redundará en una mejora de su calidad, en un mejor modelado y en resultados más fiables.
- Un entorno dinámico e interactivo que está optimizado para visualizar los datos y comprender sus relaciones.

- Ofrece uno de los conjuntos más completos de algoritmos avanzados de modelado predictivo y descriptivo, incluyendo árboles de decisión, splines de regresión, redes neuronales, regresión lineal y logística, regresión por mínimos cuadrados parciales, y muchos más. También se incluyen modelos específicos de la industria tales como la puntuación de crédito y ratemaking para el seguro.

En resumen, se trata de una de las herramientas con más potencia del mercado desde el punto de vista de trabajar con grandes bases de datos; sin embargo, contrasta con el alto precio que se ha de pagar por su licencia.

Imagen 3: Resultado de aplicar un análisis Cluster con SAS Enterprise Miner

Para obtener más información de esta herramienta se puede acceder a través del siguiente enlace: http://www.sas.com/technologies/analytics/datamining/miner/

2.5. Salford Systems Data Mining

Salford Systems es una empresa especializada, entre otras tareas, en la elaboración de software de minería de datos y consultoría. A este respecto ofrece los siguientes productos:

- *Software CART*: ofrece una clasificación multi-plataforma robusta, con una amplia variedad de análisis de alta precisión de minería de datos. Es la única herramienta basada en árboles de decisión según la metodología desarrollada por la Universidad de Stanford y la Universidad de Berkeley en California.

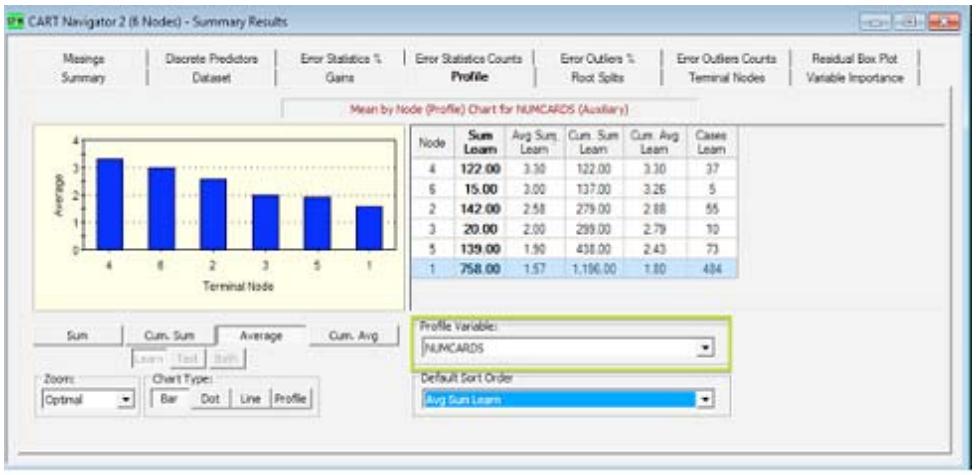


Imagen 4: Ventana de resultados en CART

- *TreeNet*: Basada en árboles de decisiones impulsadas. TreeNet es un sistema de aproximación de funciones y que también sirve como herramienta de exploración inicial de los datos.
- *RandomForests*: Ofrece modelos predictivos de alto rendimiento e incorpora nuevos análisis de clúster de métrica libre.
- *SPM Salford Predictive Modeler*: Cuenta con características adicionales orientadas a mejorar los modelos predictivos.

Para utilizar cada uno de estos programas se ha de pagar su correspondiente licencia.

Finalmente para obtener información sobre cada uno de ellos se puede consultar la web de Salford Systems a través del enlace: http://www.salford-systems.com/

2.6. Oracle Data Mining

Oracle Data Mining (ODM) es una herramienta de software desarrollada por la empresa Oracle para aplicar técnicas de minería de datos a grandes volúmenes de datos.

A través de esta herramienta se realizará el proceso de importación de los datos, su preparación, así como el desarrollo y despliegue del modelo.

La herramienta ODM está basada en un esquema de flujo de trabajo, similar a otras herramientas de minería de datos, siendo una extensión del SQLDeveloper, permitiendo analizar los datos, explorar los datos, construir y evaluar modelos y aplicar estos modelos a nuevos datos, así como compartir estos modelos en aplicaciones en línea entregando resultados en tiempo real. La herramienta integra todas las etapas

del proceso de la minería de datos y permite integrar los modelos en otras aplicaciones con objetivos similares.

ODM funciona dentro de la base de datos de Oracle, así que no hay necesidad de exportar los archivos a un paquete de software estadístico fuera de la base de datos, lo que reduce los costos y mejora la eficiencia. Con un lenguaje de procedimiento integrado/ lenguaje de consulta estructurado (PL / SQL) e interfaces de Java de programación de aplicaciones (API), Oracle DM permite a los usuarios construir modelos.

ODM ofrece dos versiones, una en la que a través de una interfaz gráfica los usuarios podrán aplicar las técnicas de minerías de datos que consideren necesarias y una versión en la que los desarrolladores podrán utilizar la API de SQP para crear aplicaciones a medida.

Imagen 5: Ventana principal de Oracle Data Miner

Se trata de la herramienta más potente para trabajar con bases de datos de Oracle, si bien habrá que pagar una licencia por su uso.

Para obtener más información sobre la herramienta se puede consultar su sección dentro de la web de Oracle a través del siguiente enlace: http://www.oracle.com/products/database/options/advanced-analytics/index.html

2.7. Rapid Miner

Esta herramienta forma parte del proyecto *Rapid-i*. Este proyecto nació en 2006 como Spin-Off de la Universidad de Dortmund, donde se inauguró la primera versión del software en 2001.

Rapid-i cuenta con dos componentes:

- *RapidMiner*: Versión stand-alone para analistas. Implementa todos los operadores de data mining, modelos predictivos, modelos descriptivos, transformación de datos, series de tiempo, etc.
- *RapidAnalytics*: Versión Servidor de RapidMiner. Permite trabajo colaborativo, escalable y concurrente de múltiples usuarios, capacidad de delegar en bases de datos (In-Database Mining) y otras mejoras de funcionalidad como: plataforma Web de publicación de informes, implementación de sistemas de scoring, diseño y navegación Web de informes, Single-sign on e integración vía Servicios Web, entre otras.

RapidMiner permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico.

Entre las características principales de *RapidMiner* destacamos que:

- Está desarrollado en Java.
- Es multiplataforma.
- Representación interna de los procesos de análisis de datos en ficheros XML.
- Permite a los experimentos componerse de un gran número de operadores anidables arbitrariamente, que se detallan en archivos XML.
- Permite el desarrollo de programas a través de un lenguaje de script.
- Puede usarse de diversas maneras:
    - A través de un GUI.
    - En línea de comandos.
    - En batch (lotes)
    - Desde otros programas, a través de llamadas a sus bibliotecas.
- Extensible.
- Incluye gráficos y herramientas de visualización de datos.
- Dispone de un módulo de integración con R.
- Software de código abierto.

Además, esta aplicación ofrece más de 500 operadores para todos los principales procedimientos de máquina de aprendizaje, y también combina esquemas de aprendizaje y evaluadores de atributos del entorno de aprendizaje Weka.

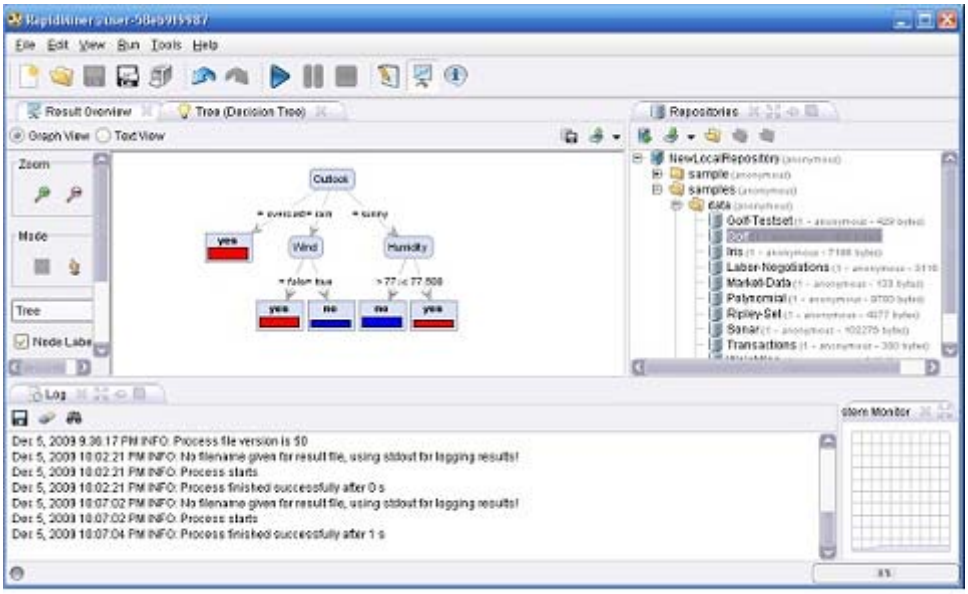


Imagen 6: Árboles de decisión con *Rapidminer*

Finalmente, al tratarse de un software libre y de código abierto puede ser descargado a través del siguiente enlace: http://rapid-i.com/content/view/181/190/

2.8. KNIME

KNIME (Konstanz Information Miner) es una plataforma de código abierto de fácil uso y comprensible para integración de datos, procesamiento, análisis y exploración.

Esta herramienta fue desarrollada originalmente en el departamento de Bioinformática y Minería de Datos de la Universidad de Constanza, Alemania, bajo la supervisión del profesor Michael Berthold. En la actualidad, la empresa KNIME.com, radicada en Zúrich, Suiza, continúa su desarrollo, además de prestar servicios de formación y consultoría.

KNIME ofrece a los usuarios la capacidad de crear de forma visual flujos o tuberías de datos, ejecutar selectivamente algunos o todos los pasos de análisis, y luego estudiar los resultados, modelos y vistas interactivas.

Está desarrollado sobre la plataforma Eclipse y programado, esencialmente, en Java. Como otros entornos de este tipo, su uso se basa en el diseño de un flujo de ejecución que plasme las distintas etapas de un proyecto de minería de datos.

Para ello, *KNIME* proporciona distintos nodos agrupados en fichas, como por ejemplo:

a) Entrada de datos [IO > Read]
b) Salida de datos [IO > Write]

c) Preprocesamiento [Data Manipulation], para filtrar, discretizar, normalizar, filtrar, seleccionar variables, etc.
d) Minería de datos [Mining], para construir modelos (reglas de asociación, clustering, clasificación, MDS, PCA…)
e) Salida de resultados [Data Views] para mostrar resultados en pantalla (ya sea de forma textual o gráfica)

Por otro lado, a través de plugins, los usuarios pueden añadir módulos de texto, imágenes, procesamiento de series de tiempo y la integración de varios proyectos de código abierto, tales como el lenguaje de programación *R*, *WEKA*, el kit de desarrollo de Química y LIBSVM.

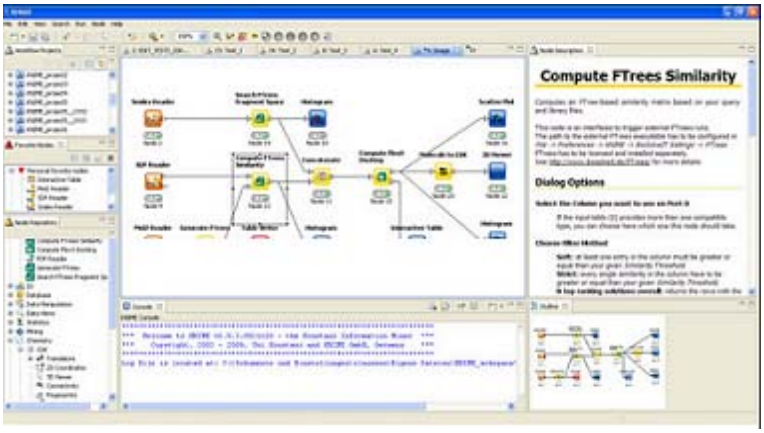


Imagen 7: Árboles de decisión con *KNIME*

Finalmente se trata de una herramienta multiplataforma que puede ser descargada junto con su documentación a través del enlace http://www.knime.org/ .

2.9. R

R es un entorno estadístico tremendamente potente y completo. Las llamadas a *R* se realizan en línea de comando, si bien existen algunas interfaces gráficas (*Rcommander*, etc) que facilitan el uso de este programa. Fue desarrollado inicialmente por el Departamento de Estadística de la Universidad de Auckland, Nueva Zelanda, en 1993.

R es un lenguaje de programación y entorno de software de código abierto para computación y gráficos estadísticos. Proporciona múltiples técnicas para simulación, modelado lineal y no lineal, análisis de series temporales, pruebas estadísticas clásicas, clasificación, agrupación en clústeres, etc.

El entorno de *R* se caracteriza por su flexibilidad e incluye, entre otros:

- Un buen gestor de datos.
- Un conjunto de operadores para cálculos en arrays (vectores de gran tamaño)
- Un conjunto integrado de herramientas de análisis de datos.

- Funciones gráficas para análisis y visualización de los datos.
- Un lenguaje de programación simple que incluye condicionales, bucles, funciones recursivas definidas por el usuario y capacidades de entrada y salida.

En relación al proceso de minería de datos, R posee gran cantidad de paquetes estadísticos útiles para realizar este proceso; en especial, destacaremos:

- *Rattle*: que ofrece al usuario una interfaz gráfica para aplicar técnicas de minería de datos a grandes bases de datos.

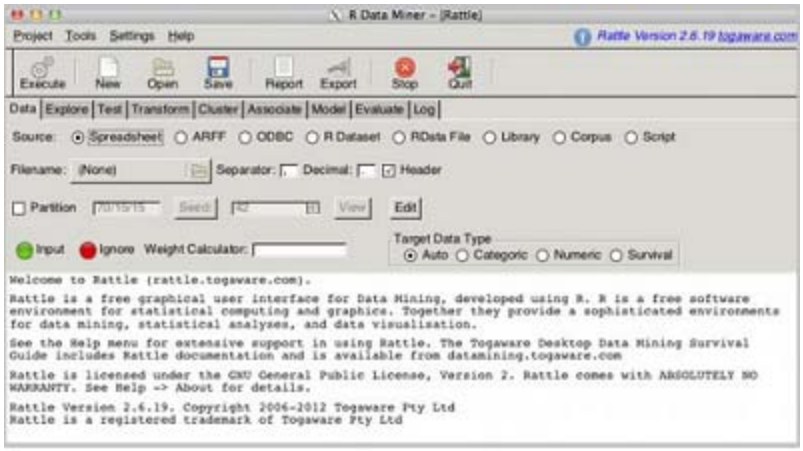


Imagen 8: Interfaz de *Rattle*

- *Caret*: que, más allá de integrar diversos algoritmos, incluye funciones auxiliares útiles para seleccionar modelos, comparar la importancia de funciones, realizar validaciones cruzadas, etc., utilizando una sintaxis coherente y homogénea.

- *RDataMining*. El objetivo de sus promotores es incluir en él algoritmos publicados que no tengan todavía implementación en R.

Se trata de un software libre, distribuido bajo licencia GPL, muy extendido en la comunidad universitaria y que está llamado a cobrar un papel cada vez más relevante en el mundo de las aplicaciones profesionales y de la empresa.

Tanto el programa como los paquetes estadísticos y su documentación asociada pueden descargarse a través de la web del proyecto R: http://www.r-project.org/ .

2.10. Orange

Se trata de una suite para minería de datos y aprendizaje automático, desarrollado en la Facultad de Informática de la Universidad de Ljubljana (Eslovenia)

Esta herramienta cuenta con un fácil y potente, rápido y versátil front-end de programación visual para el análisis exploratorio de datos y visualización, y librerias para Python y secuencias de comando.

Contiene un completo juego de componentes desarrollados en C++. para preprocesamiento de datos, características de puntuación y filtrado, modelado, evaluación del modelo y técnicas de exploración. A estos componentes se puede acceder de dos formas:

- Por medio de scripts desde Python.
- Por medio de widgets (componentes GUI), desde CANVAS.

Se trata de una aplicación multiplataforma y se distribuye bajo licencia GPL.

Además, *orange* proporciona componentes para:

- Entrada/salida de datos, soportando los formatos C4.5, assistant, retis y tab (nativo)
- Preprocesamiento de datos: selección, discretización, etc.
- Modelado predictivo: árboles de clasificación, regresión logística, clasificador de Bayes, reglas de asociación, etc.
- Métodos de descripción de los datos: mapas autoorganizados, k-means clustering, etc.
- Técnicas de validación del modelo, como la validación cruzada.

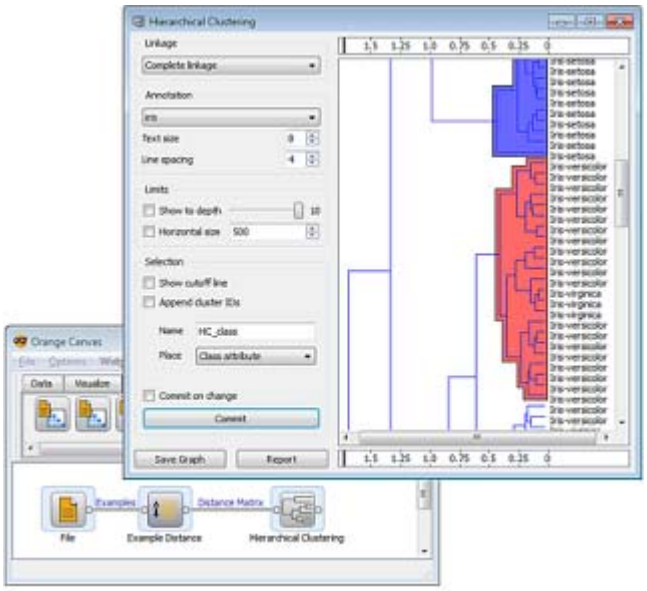


Imagen 9: Flujo de trabajo y Clustering con *Orange*

Finalmente, si se desea descargar y conocer más en profundidad la herramienta es recomendable visitar la página web de sus creadores: http://orange.biolab.si/.

2.11. WEKA

WEKA, acrónimo de Waikato Environment for Knowledge Analysis, es un conjunto de librerías JAVA para la extracción de conocimiento desde bases de datos. Está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación y visualización.

Se trata de un software desarrollado en la Universidad de Waikato (Nueva Zelanda) bajo licencia GNU-GPL lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años. Se trata de una herramienta de gran potencia, si bien no tiene implementados, a fecha de hoy, algoritmos para la realización de un modelado de secuencias.

Tanto la aplicación como los manuales de referencia se pueden descargar a través de la web del proyecto: http://www.cs.waikato.ac.nz/ml/weka/.

Dado que este es el programa que utilizaremos en nuestro trabajo, pasamos a estudiarlo en profundidad en el siguiente apartado.

3. Introducción a WEKA

3.1. Introducción

WEKA, acrónimo de Waikato Environment for Knowledge Analysis, es un entorno para experimentación de análisis de datos que permite aplicar, analizar y evaluar las técnicas más relevantes de análisis de datos, principalmente las provenientes del aprendizaje automático, sobre cualquier conjunto de datos del usuario.

Este programa se distribuye como software de libre distribución (licencia GNU-GPL) desarrollado en Java y dispone de tres entornos de trabajo gráficos y un entorno en modo consola, permitiendo la implementación de algoritmos para preprocesamiento de datos, clasificación, regresión, clustering, selección de atributos, reglas de asociación, etc.

El desarrollo de WEKA se inició en 1993 en la Universidad de Waikato (Nueva Zelanda) siendo la primera versión pública Weka 2.1 la del año 1996. Actualmente, la última versión de WEKA es la 3.6 estando disponible para los principales sistemas operativos tanto libres como comerciales.

WEKA se puede descargar en la web de la Universidad de Waikato a través del enlace http://www.cs.waikato.ac.nz/ml/weka/. Se puede consultar desde el manual de referencia para la aplicación y otras publicaciones relacionadas, como descargar ejemplos para realizar ensayos con esta herramienta:

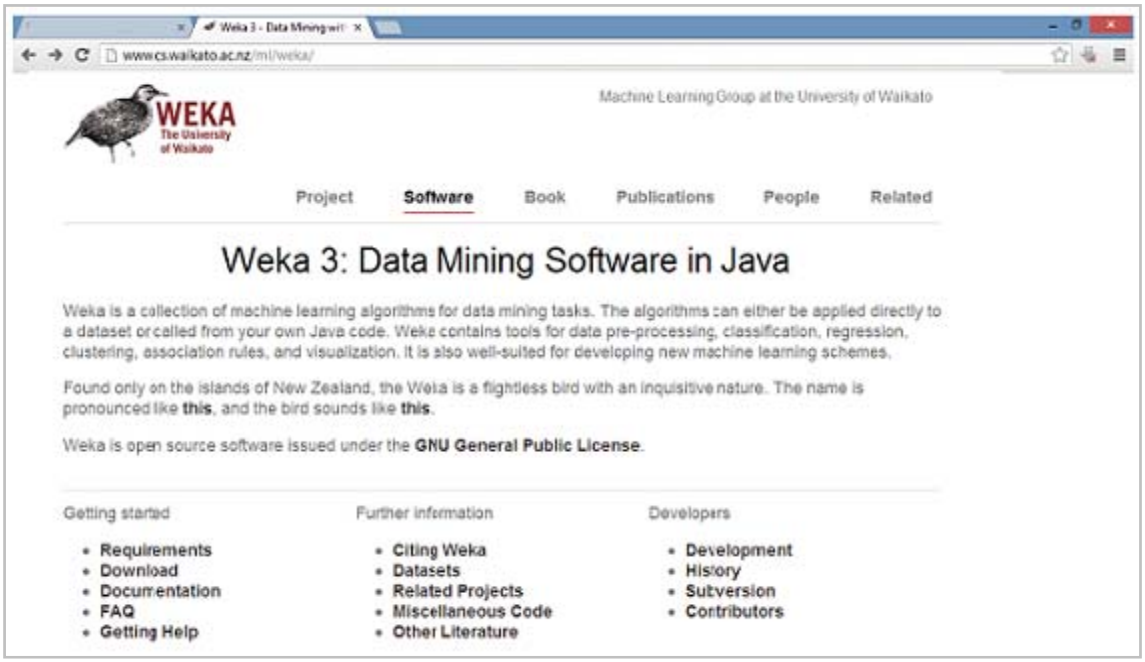


Imagen 10: Web de WEKA

El potencial usuario podrá descargar e instalar WEKA siguiendo las directrices del manual de referencia. Una vez la instalación se haya realizado de forma correcta podrá acceder a la herramienta cuya interfaz de inicio se muestra a continuación:

Imagen 11: Interfaz principal de WEKA

A través de la intefaz se puede acceder a las distintas aplicaciones que componen WEKA; esto es, Simple Cli, Explorer, Experimenter y Knowledge Flow. Todas estas herramientas se explicarán en los siguientes puntos de este capítulo.

3.2. Los datos

WEKA utiliza un formato de datos denominado arff (Attribute Relation File Format). Cada uno de estos ficheros consta de 3 partes:

- Cabecera. Definida por: @relation <nombre-conjunto-de-datos>
- Declaración de atributos o variables. A través de:

  @attribute <nombre-variable> <tipo>

  siendo el valor de *tipo*: string, numeric, integer, date o nominal.
- Sección de datos. Definidos de la siguiente forma:

  @data

  donde se tendrá una línea para cada registro, los valores estarán separados por comas y los valores perdidos se representan mediante el carácter *?*.

Además es posible escribir comentarios en ese fichero, precedidos del caracter %.

```
% Fichero de clientes de un gimnasio en formato arff
@relation clientes
@attribute nombre STRING
@attribute sexo {hombre,mujer}
@attribute edad INTEGER
@attribute peso NUMERIC
@attribute ingreso DATE "dd-MM-yyyy HH:mm"
@data
Rafael,hombre,45,75.6,"12-04-2012 07:20"
" Fernando",hombre,?,57.5, "04-01-2012 12:05"
Carmen,mujer,43,?, " 20-02-2012 09:14"
Luisa,mujer,?,?, "15-03-2012 10:00"
```

Imagen 12: Ejemplo de fichero de datos arff

3.3. Simple CLI

Se trata de una de las aplicaciones a la que se puede acceder a través de la interfaz principal de WEKA.

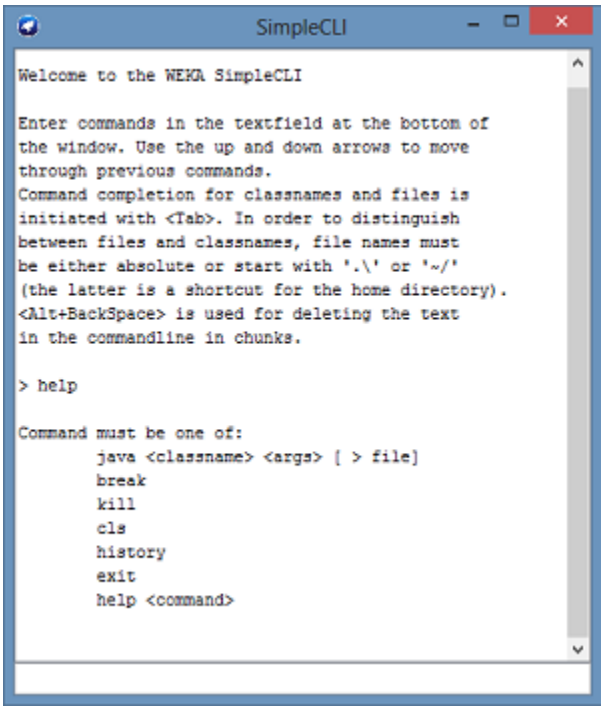


Imagen 13: Interfaz de Simple CLI

Simple CLI es la abreviatura de Simple Command-Line Interface (Interfaz Simple de Línea de Comandos) y se define como una ventana de comandos java. Nace con la primera versión de WEKA y a través de ella se ejecutan las clases allí implementadas. Hoy día y debido a la aparición de las otras aplicaciones que componen WEKA es menos utilizada, ya que estas nuevas herramientas constan de interfaces gráficas que facilitan su uso por parte del usuario.

En la interfaz se pueden ejecutar los siguientes comandos:

- java <nombre-de-la-clase><args>: Permite ejecutar una determinada clase de WEKA.
- break: Detiene la tarea actual.

- kill: Finaliza la tarea actual.
- cls: Limpia el contenido de la consola.
- history: Muestra el historial de ordenes ejecutadas.
- exit: Sale de la aplicación.
- help <comando>: Proporciona una breve descripción de cada mandato.

3.4. Explorer

Se trata de otra de las aplicaciones a la que se accede a través de la interfaz principal de WEKA. Esta herramienta permite, entre otras tareas, llevar a cabo la ejecución de los algoritmos de análisis implementados sobre los ficheros de entrada. A estas funcionalidades se puede acceder a través de las siguientes pestañas:

- Preprocess: permite la visualización y preprocesado de los datos (aplicación de filtros)
- Classify: útil para la aplicación de algoritmos de clasificación y regresión.
- Cluster: conjunto de técnicas de agrupación.
- Associate: métodos de asociación.
- Select Attributes: selección de atributos.
- Visualize: visualización de los datos por parejas de atributos.

Al entrar en la aplicación, la interfaz aparecerá vacía, mientras que la pestaña Preprocess está seleccionada.

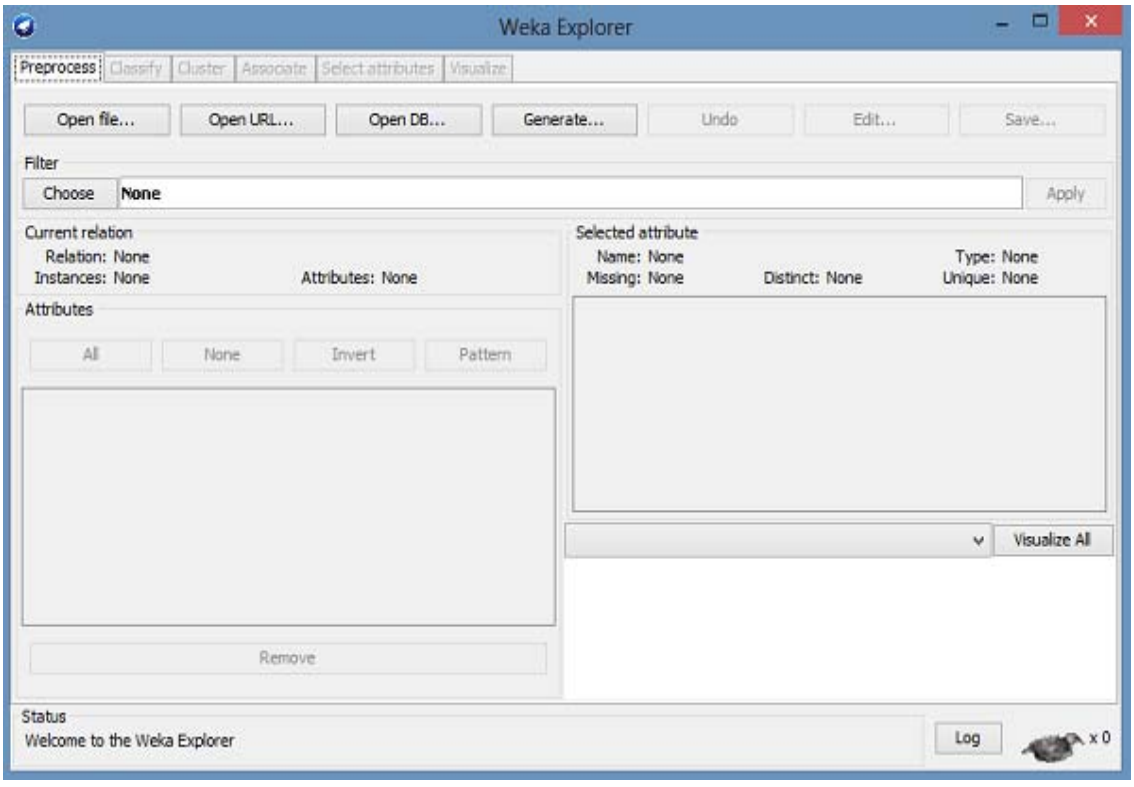


Imagen 14: Interfaz de Explorer con la pestaña Preprocess activada

3.4.1. Pestaña Preprocces

Esta es la primera pestaña de la aplicación y es imprescindible para realizar cualquier tipo de análisis pues es en ella donde introduciremos los datos con los que vamos a trabajar, bien a través de un fichero (Open file), bien a través de una url (Open URL), bases de datos (Open Database) o bien introduciendo la información directamente en la aplicación (Generate)

Si en la web de WEKA hemos descargado previamente el fichero *weather.arff*, ahora podremos cargarlo pulsando el botón *Open file*, siendo el resultado de ello el que se muestra en la siguiente imagen.

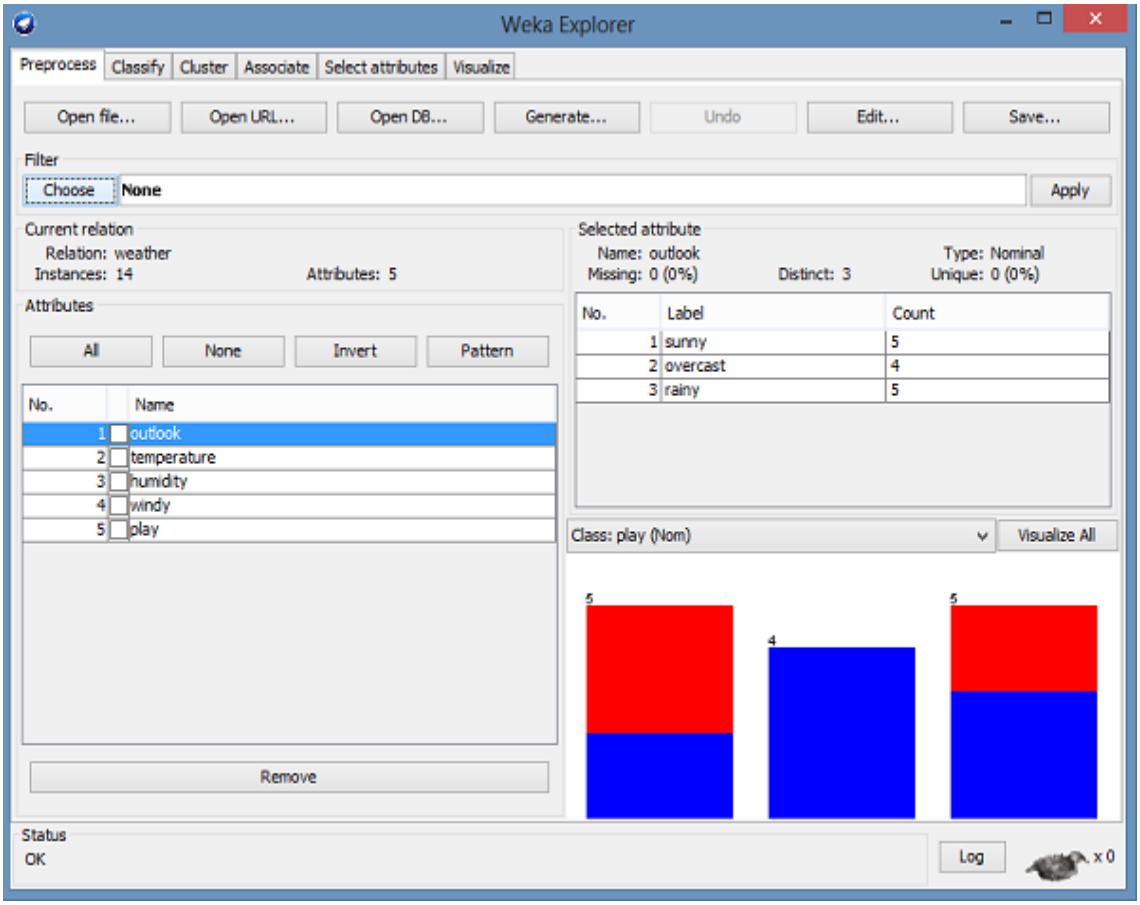


Imagen 15: Pestaña Preprocess con el fichero weather.arff cargado

Una vez cargado el fichero se visualizan los atributos o variables del fichero y para aquellos que seleccionemos (*Attributes*) podremos ver un resumen estadístico (*Selected attribute*)

Por otro lado las herramientas de preprocesamiento se denominan filtro (*Filter*) y cada filtro actúa en uno de los siguientes niveles:

- Atributos: Actúan "en vertical" en la base de datos, modificando un atributo completo. Ejemplo: Filtro de discretización.
- Instancias: Actúan en horizontal, seleccionando un grupo de registros (instancias). Ejemplo: Filtro de selección aleatoria.

Para visualizar los filtros bastaría pulsar el botón *Choose* dentro de la sección *Filter*. Es ahí donde seleccionaremos que filtro utilizar en función de nuestro objetivo. Los filtros se dividen en:

- No supervisados: en su funcionamiento no interviene ningún algoritmo externo.
- Supervisados: actúan en conjunción con clasificadores para analizar su efecto y guiar su actuación.

Por ejemplo, el filtro no supervisado Discretice divide el recorrido del atributo numérico en intervalos, que pueden ser de la misma amplitud o con el mismo número de observaciones (aproximadamente). Además se crea un atributo nominal en el que cada categoría corresponde a un intervalo.

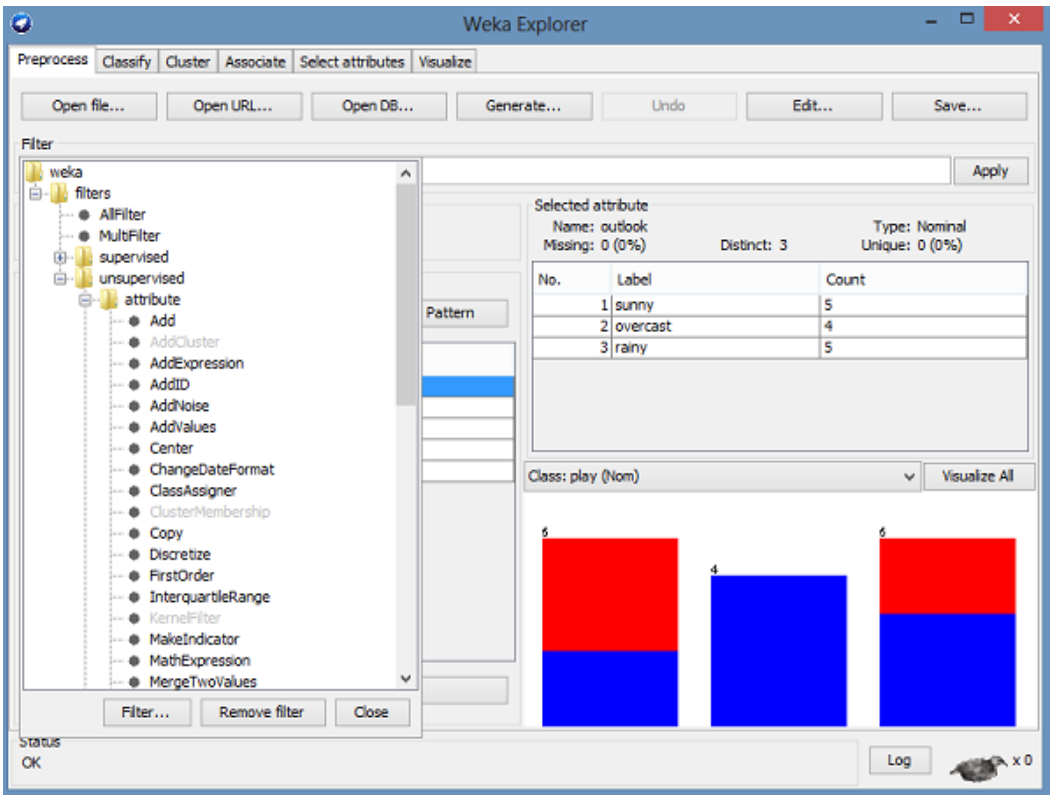


Imagen 16: Selección del tipo de filtro en la pestaña Preprocess

3.4.2. Pestaña Classify

En esta pestaña se podrá definir y resolver un problema de clasificación. Puede ocurrir que, en ocasiones, el problema de clasificación se formule como un refinamiento en el análisis, una vez que se han aplicado algoritmos no supervisados de agrupamiento y asociación para describir relaciones de interés en los datos.

Además se busca construir un modelo que permita predecir la categoría de las instancias en función de una serie de atributos de entrada. En el caso de WEKA, la clase es simplemente uno de los atributos simbólicos disponibles, que se convierte en

la variable objetivo a predecir. Por defecto, es el último atributo (última columna) a no ser que se indique otro explícitamente.

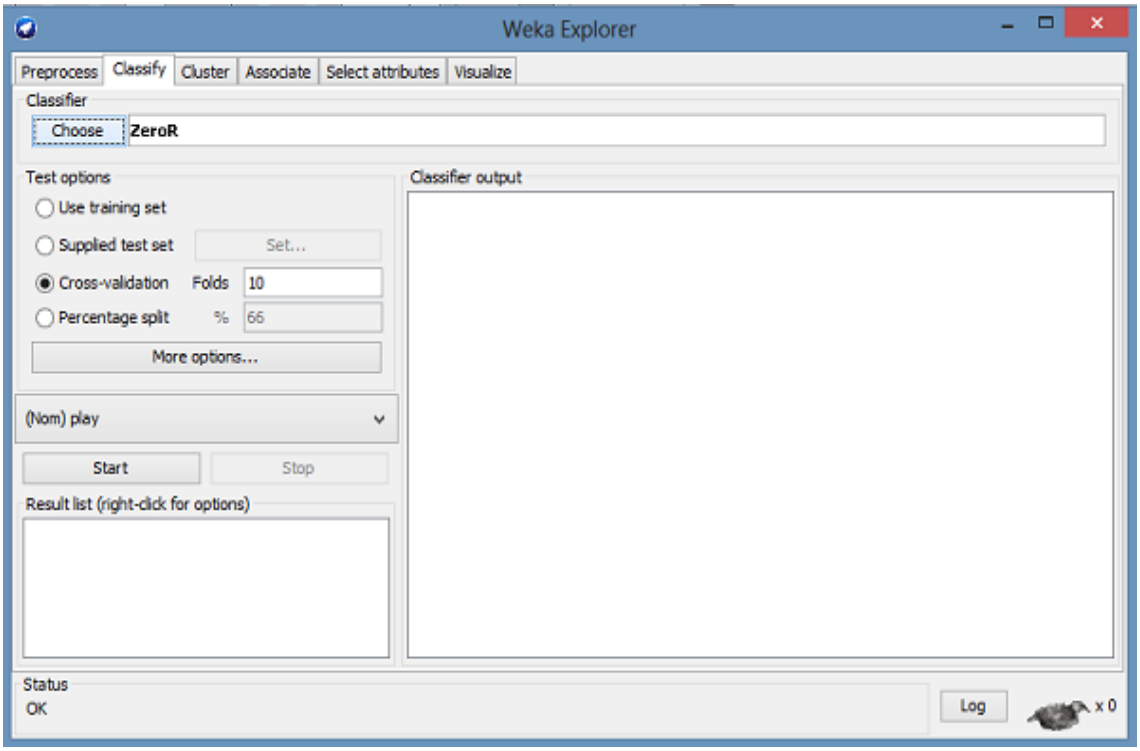


Imagen 17: Pestaña Classify

Al pulsar sobre la pestaña aparece por defecto el clasificador *ZeroR*, si bien podrá seleccionarse otro pulsando el botón *Choose*.

En la parte inferior (*Test options*) se podrán incluir y modificar los parámetros asociados al clasificador a través de las opciones:

- *Use training set*: En esta opción se entrenará el método con todos los datos disponibles y luego se aplicará sobre los mismos.
- *Supplied test set*: Marcando esta opción tendremos la oportunidad de seleccionar un fichero de datos con el que se probará el clasificador obtenido con el método de clasificación usado y los datos iniciales.
- *Cross-validation*: La herramienta realizará una validación cruzada estratificada del número de particiones dado (Folds).
- *Percentage split*: Se define un porcentaje de los datos con el que se construirá el clasificador y con la parte restante se realizarán las pruebas.

Para introducir más opciones (*Output Model, Output per-class stats, Output entropy evaluation mesures, Output confusion matrix*) pulsaríamos el botón *More options*. Finalmente y pulsando el botón *Start* se podrán visualizar los resultados en la sección *Classifier output* en función del que hayamos seleccionado en la sección *Result list*.

3.4.3. Pestaña Cluster

WEKA ofrece distintas posibilidades de aplicar algoritmos de *clustering* (o clasificación no supervisada) sobre los datos. Las técnicas de *clustering* se utilizan en bases de datos no supervisadas en las que la variable clase no existe (o no se ha definido). Así pues el objetivo fundamental de esta técnica es descubrir (*class discovery*) dichas clases o estructuras diferenciadas en los datos de estudio. El programa WEKA ofrece en la pestaña Cluster varios algoritmos de cluster, entre ellos el de K-medias y el EM.

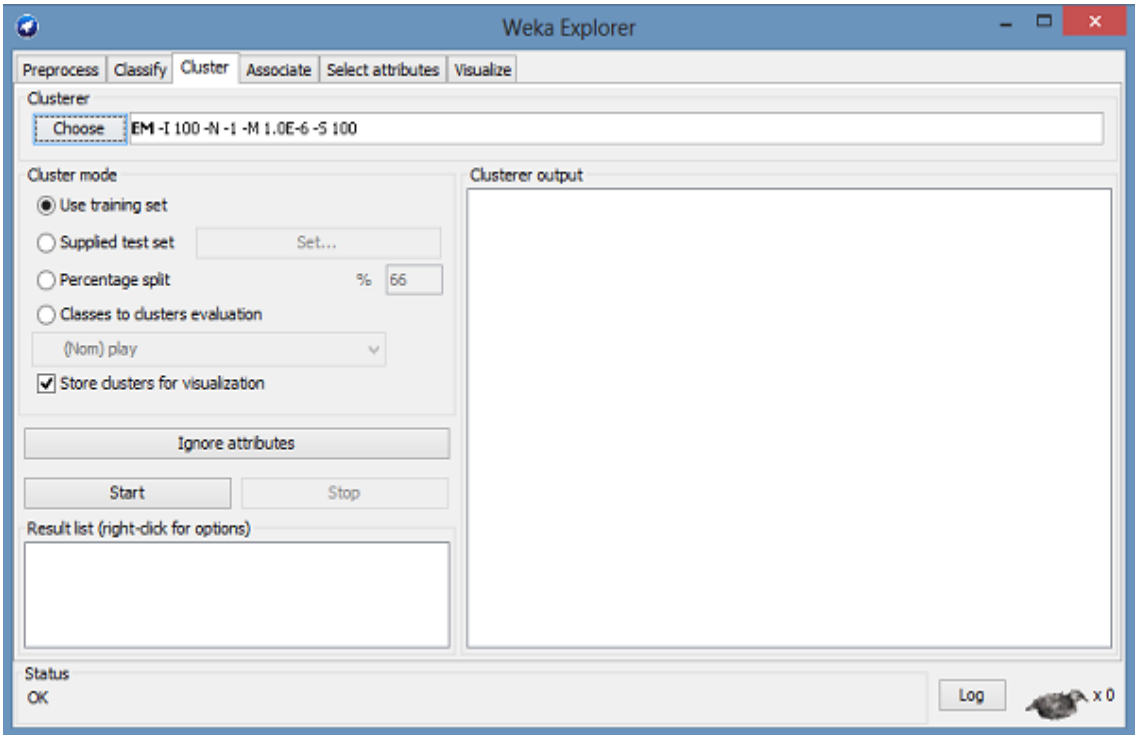


Imagen 18: Pestaña Cluster

La distribución de las secciones y opciones de la sección *Cluster* es muy similar al de la pestaña *Classify*. Por lo tanto, una vez elegido el método de *clustering*, se seleccionan las opciones pertinentes, con el botón *Start* se ejecuta el proceso y en las secciones *Result list* y *Clusterer ouput* se visualizarán los resultados.

3.4.4. Pestaña Associate

En esta pestaña el usuario podrá realizar diferentes algoritmos de asociación. Estos algoritmos permiten la búsqueda automática de reglas que relacionan conjuntos de atributos entre sí. Son algoritmos no supervisados, en el sentido de que no existen relaciones conocidas a priori con las que contrastar la validez de los resultados, sino que se evalúa si esas reglas son estadísticamente significativas.

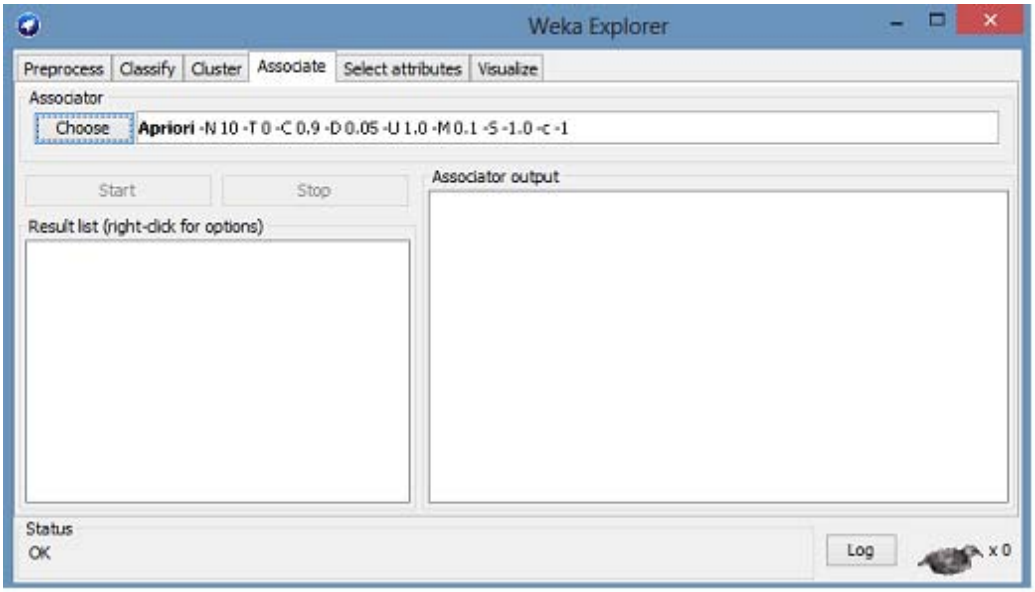


Imagen 19: Pestaña Associate

Pulsando el botón *Choose* elegiremos el algoritmos de asociación que deseemos y pulsando *Start* obtendremos los resultados que se podrán visualizar a través de las secciones *Result list* y *Associator output*.

3.4.5. Pestaña Select attributes

En esta pestaña trataremos de determinar qué atributos formarán parte del modelo; es decir, eliminaremos aquellos atributos que resulten redundantes e irrelevantes. Además, si hay un número excesivo de atributos puede conllevar a obtener un modelo demasiado complejo y se produzca sobreajuste.

En WEKA, la selección de atributos se puede hacer de varias maneras, siendo la más directa la que se realiza a través de esta pestaña. En ella tenemos que seleccionar:

- El método de evaluación (*Attribute Evaluator*): es la función que determina la calidad del conjunto de atributos para discriminar la clase. Se pueden distinguir entre los métodos que directamente utilizan un clasificador específico para medir la calidad del subconjunto de atributos a través de la tasa de error del clasificador y los que no.
- El método de búsqueda (*Search Method*): es la manera de realizar la búsqueda de conjuntos de forma eficiente.

Una vez seleccionado alguno de estos métodos podremos determinar la forma en que seleccionaran los atributos (usando un conjunto completo de entrenamiento o mediante validación cruzada) en la sección *Attribute Selection Mode*.

Finalmente los resultados obtenidos se podrán visualizar a través de las secciones *Result list* y *Associator output*.

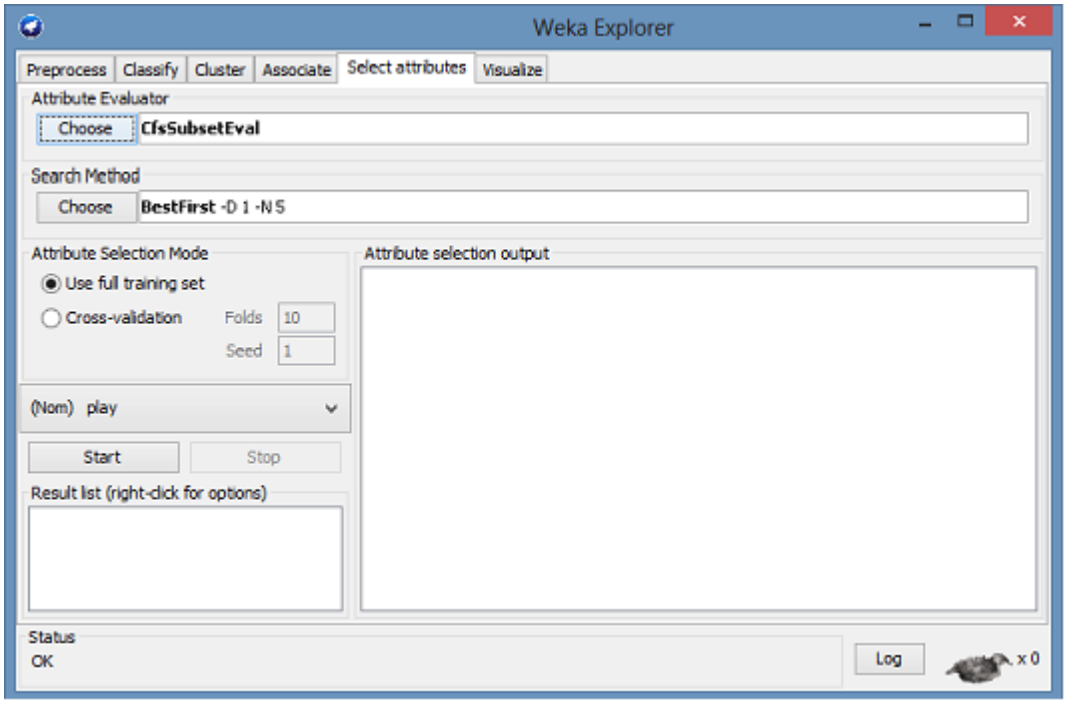


Imagen 20: Pestaña Select attributes

3.4.6. Pestaña Visualize

En último lugar, aunque su utilización puede ser recomendable en las primeras etapas del proceso de análisis, se encuentra la pestaña *Visualize*.

La herramienta de visualización de WEKA permite presentar gráficos 2D interactivos que relacionen pares de atributos, con la opción de utilizar además los colores para añadir información de un tercer atributo. Además, permite detectar gráficamente la existencia de asociaciones y correlaciones entre atributos, así como seleccionar instancias de forma gráfica, que pueden ser almacenadas posteriormente en formato *arff*.

Al pulsar sobre la pestaña, aparecerán los gráficos correspondientes a todas las combinaciones posibles de atributos. Debajo de ellos aparecen varias opciones de edición de gráficos:

- *Plot size:* indica el tamaño del gráfico en píxeles.
- *Point Size:* define el tamaño del punto en píxeles.
- *Jitter:* Crea un ruido aleatorio a las muestras, de manera que espacia las muestras que están físicamente muy próximas, esto tiene útilidad cuando se concentran tanto los puntos que no es posible discernir la cantidad de éstos en un área.
- *Color*: Indica los colores que se utilizarán para las clases de los atributos.

Podremos visualizar los cambios efectuados en el diseño de los gráficos pulsando *Update*, modificar el número de atributos que se van a representan eligiéndolos a través del botón *Select attributes* e indicar el tamaño de la submuestra (*SubSample*)

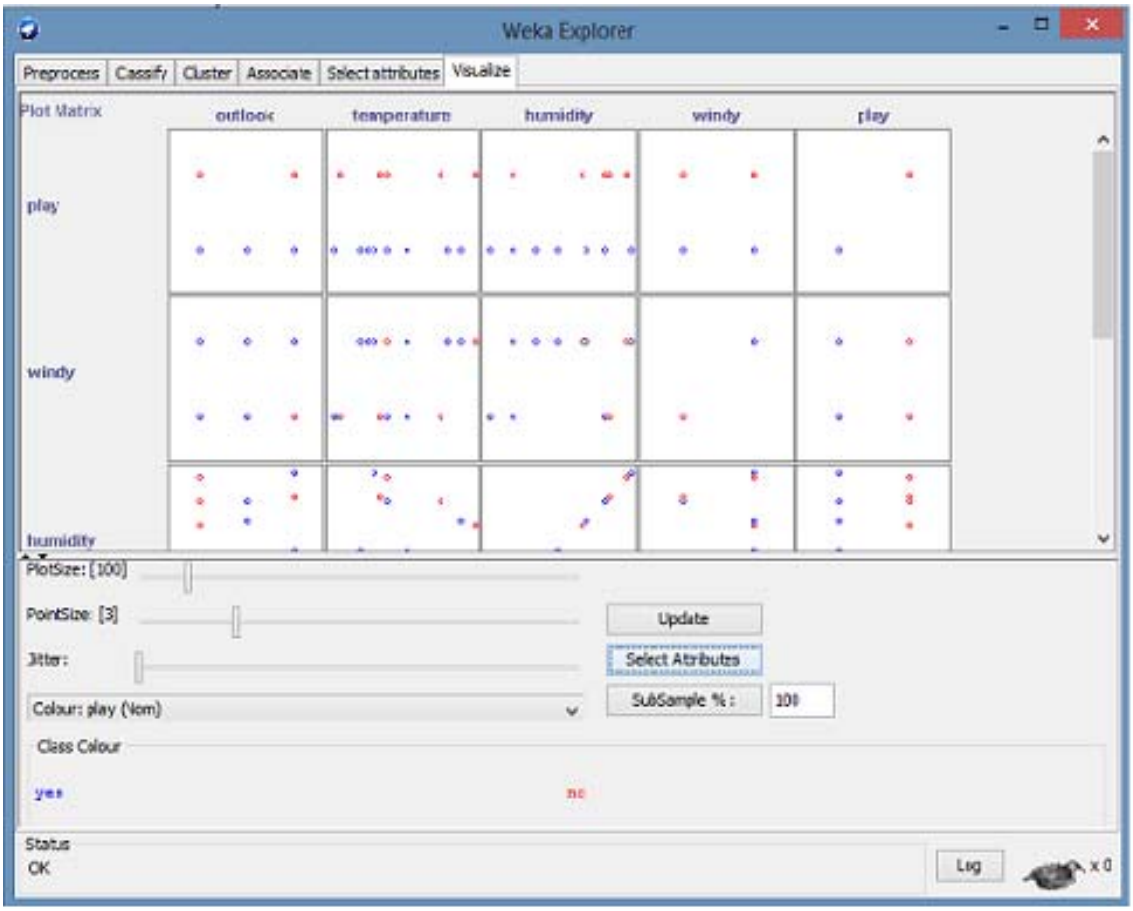


Imagen 21: Pestaña Visualize

Finalmente pulsando sobre cada gráfico podemos verlo ampliado y modificar sus atributos a través de las opciones que existen al respecto, así como guardar en un fichero *arff* los datos allí visualizados.

3.5. Experimenter

Se trata de otra de las herramientas que aparecen en la interfaz inicial de WEKA y con ella trataremos de comparar el rendimiento de los distintos algoritmos implementados en la aplicación. Además permite aplicar diferentes algoritmos sobre diferentes conjuntos de datos, lo que resulta muy útil para realizar contrastes de hipótesis o elaborar indicadores estadísticos; resultando de gran importancia en problemas de clasificación y regresión.

Al acceder a esta herramienta se visualizan, en la parte superior, tres pestañas que analizamos a continuación:

3.5.1. Pestaña Setup

En ella, Experimenter se puede configurar de dos formas, *Simple* o *Advanced*:

- **Configuración *Simple*:**

Es la configuración que aparece por defecto cuando se pulsa la pestaña Experimenter. En ella habrá que definir un fichero configuración que contendrá todos los ajustes, ficheros involucrados, notas, etc, pertenecientes al experimento y un fichero de salida donde se guardarán los resultados.

Seguidamente en la sección Experiment Type se introducirá el tipo de validación que tendrá el experimento; esto es, validación-cruzada estratificada, entrenamiento con un porcentaje de la población tomando ese porcentaje de forma aleatoria y entrenamiento con un porcentaje de la población tomando el porcentaje de forma ordenada.

En la sección *Datasets* indicaremos qué archivos de datos formarán parte del experimento y debajo se encuentra la opción *Use relative paths*, que se utiliza para que las referencias a los ficheros sean relativas.

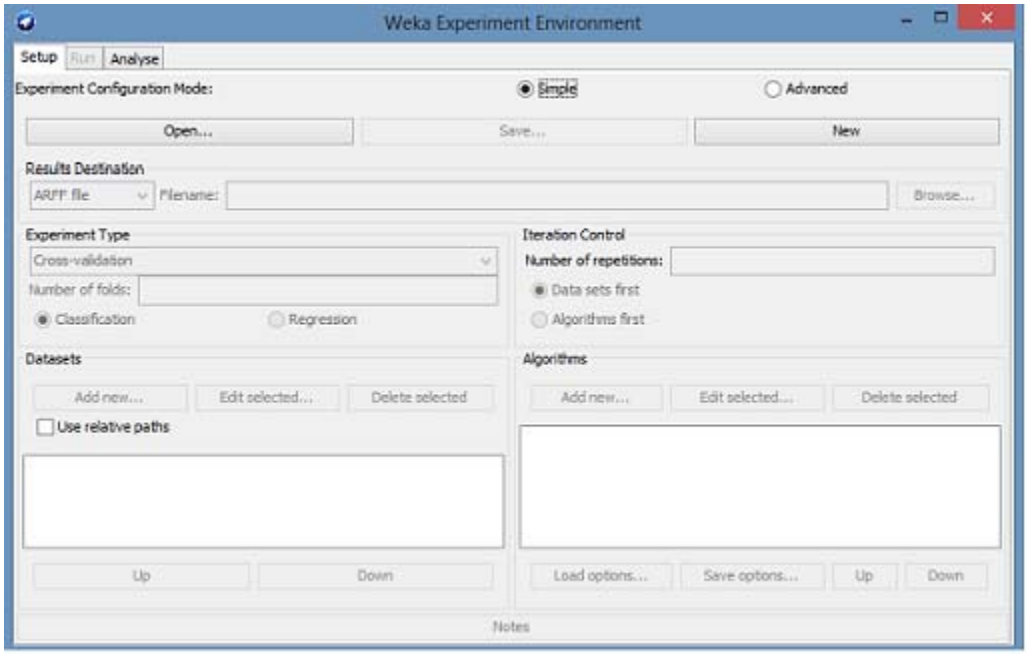


Imagen 22: Pestaña Setup con configuración Simple

En la sección *Iteration Control* se introducirá el número de repeticiones del experimento, especificando si se quiere que se realicen primero los archivos de datos o los algoritmos. Debajo de la sección *Algoritms* el usuario podrá introducir, modificar o eliminar el algoritmo que desee.

- **Configuración *Advanced*:**

Esta configuración está orientada a realizar tareas específicas más concretas que las obtenidas a través de un experimento sencillo. A través de esta opción es posible repetir el experimento variando el tamaño del conjunto de datos, distribuir la ejecución del experimento entre varios ordenadores o realizar experimentos de modo incremental.

Para comenzar a trabajar se ha de introducir el fichero configuración, un fichero de resultados e introducir y configurar el método generador de resultados que vamos a utilizar. WEKA permite los siguientes métodos:

- *CrossValidationResultProducer*: Genera resultados fruto de una validación cruzada.
- *AveragingResultProducer*: Toma los resultados de un método generador de resultados y se calculan los promedios de los resultados.
- *LearningRateResultProducer*: Llama a un método generador de resultados para ir repitiendo el experimento variando el tamaño del conjunto de datos.
- *RandomSplitResultProducer*: Crea un conjunto de entrenamiento y de prueba aleatorio para un método de clasificación dado.
- *DatabaseResultProducer*: A partir de una base de datos toma los resultados que coinciden con los obtenidos con un método generador de resultados.

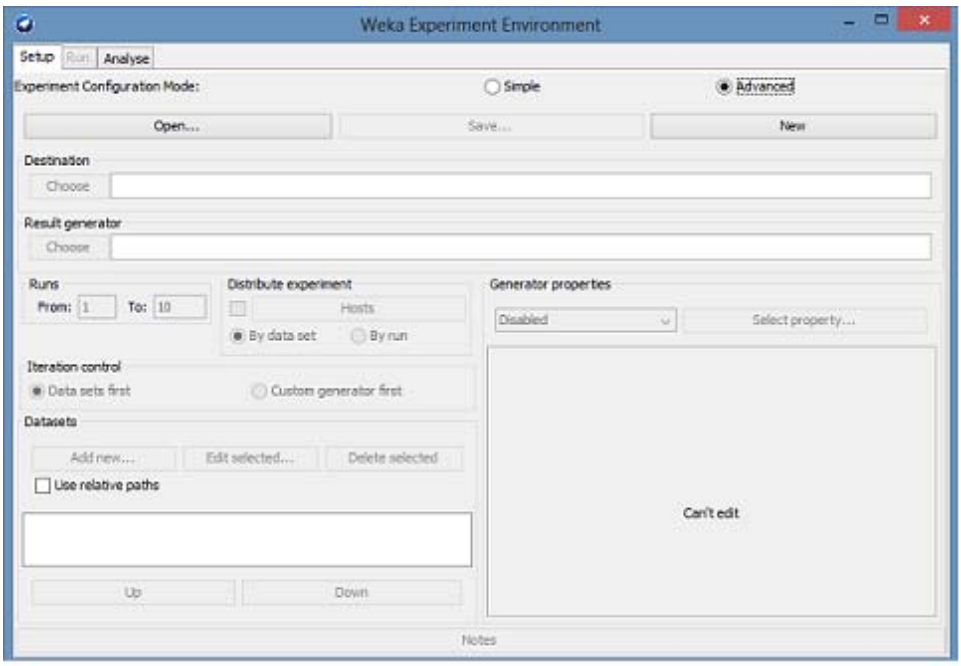


Imagen 23: Pestaña Setup con configuración Advanced.

Una vez seleccionado el generador de resultados podemos editar algunas de sus propiedades, además de añadir algoritmos, en la sección *Generator properties*, mientras que en la sección *Runs* se puede seleccionar las iteraciones con la que se realizará el experimento.

Una de las características más interesantes del modo experimentador es que permite distribuir la ejecución de un experimento entre varios ordenadores mediante Java RMI. Esta tarea se llevará a cabo en la sección *Distribute experiment*.

Otras secciones de la herramienta son *Iteration control*, que es donde se establece el orden de la iteración y *Datasets*, donde se definen los conjuntos de datos sobre los que actuarán los algoritmos de aprendizaje.

3.5.2. Pestaña Run

En esta pestaña el usuario ejecutará o detendrá el experimento. En la sección *Log* aparecerá cierta información sobre el proceso de ejecución (hora de inicio, finalización, posibles errores, etc.)

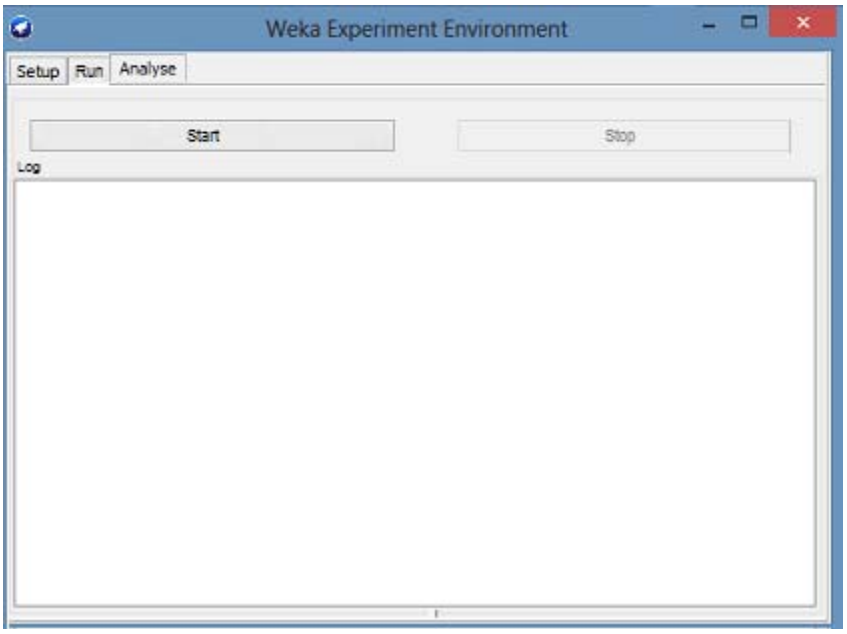


Imagen 24: Pestaña Run

3.5.3. Pestaña Analyse

En esta pestaña analizaremos los datos; es decir, podremos ver los resultados de los experimentos, realizar contrastes estadísticos, etc.

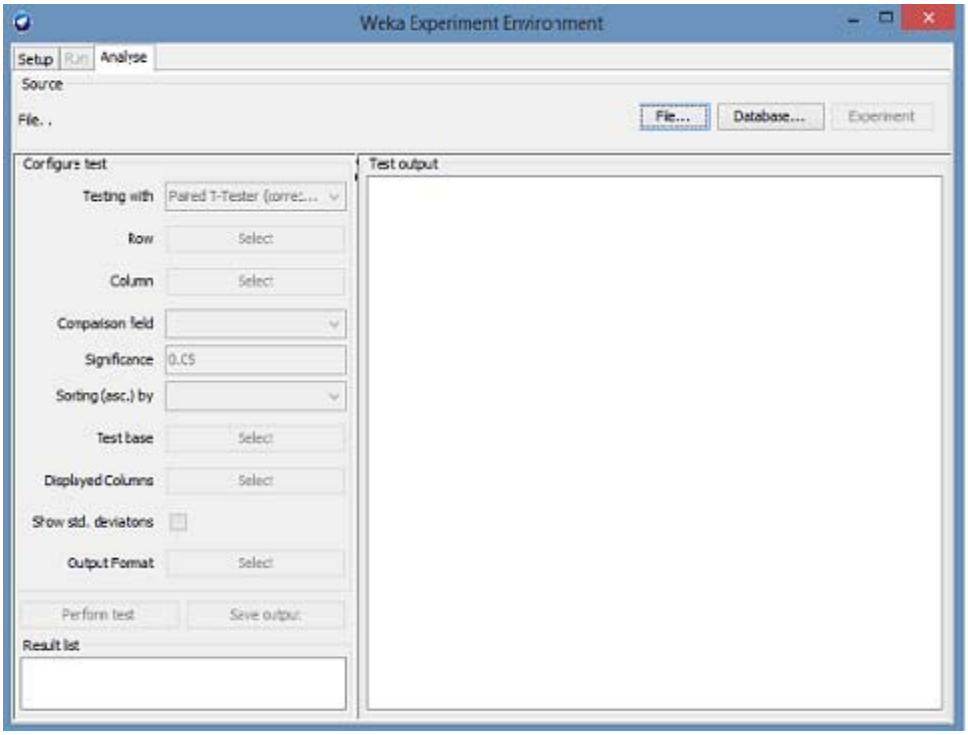


Imagen 25: Pestaña Analyse

Una vez seleccionados, en la sección Source, los datos de los resultados (pulsando *File*, *Database* o *Experiment*) se define el test que queremos realizar en la sección *Configure test*; que tendrá como opciones:

- *Testing with*: Selección del test que vamos a aplicar.
- *Row*: Define los atributos que actuarán como filas en la matriz de resultados.
- *Column*: Define los atributos que actuarán como columnas en la matriz de resultados.
- *Comparison fields*: El atributo que va a ser comparado en el contraste.
- *Significance:* Nivel de significación para realizar el contraste estadístico.
- *Sorting (asc.) by*: Obtener los resultados ordenados de forma ascendente según un atributo.
- *Test base*: Seleccionamos qué algoritmo de los utilizados se usa de base para realizar el test.
- *Displayed Columns:* Columnas que se van a mostrar.
- *Show std. Deviations:* Marcamos si queremos que se muestren las desviaciones típicas.
- *Output Format:* Si deseamos almacenar los resultados del experimento.

Finalmente pulsando *Perform test* realizaremos el experimento cuyos resultados se visualizarán en la sección *Test output*.

3.6. KnowledgeFlow

La herramienta *KnowledgeFlow* (flujo de conocimiento) muestra de una forma gráfica el desarrollo del experimento que se realiza en WEKA. Así pues se basa en situar en el panel de trabajo (*Knowledge Flow Layout*), elementos base (situados en la sección superior) de manera que creemos un circuito o flujo que defina nuestro experimento.

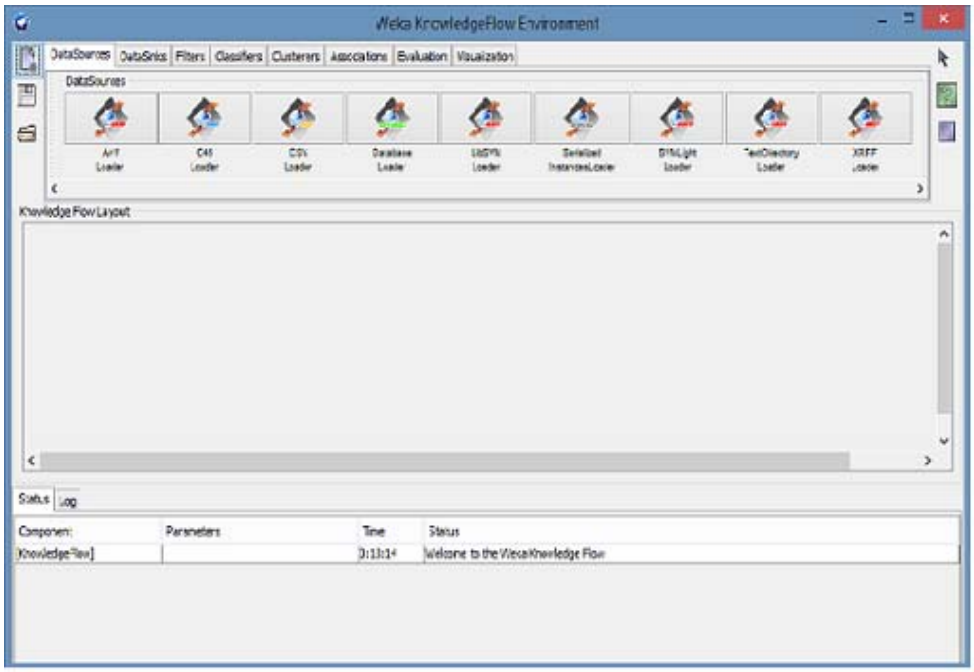


Imagen 26: Interfaz de KnowledgeFlow

En la parte superior de la pantalla aparecen un conjunto de pestañas:

- *Data Sources:* Donde elegiremos el fichero de entrada de datos. Existen varias opciones y para seleccionarlas bastaría con marcarlas con el botón izquierdo y marcar el lugar donde la vamos a situar dentro del panel de trabajo.
- *Data Sinks:* Donde elegiremos dónde se almacenarán los resultados del experimento. La forma de seleccionarlos es análoga a la anteriormente descrita.
- *Filters:* Donde definiremos los filtros que aplicaremos a los datos.
- *Classifiers:* En este caso estableceremos algoritmos de clasificación y regresión.
- *Associations:* Se aplicarán métodos de asociación a los datos.
- *Evaluation:* Donde establecer distintas técnicas de evaluación de los resultados.
- *Visualize:* Se podrán realizar los resultados y los gráficos asociados al experimento.

Para añadir las especificaciones de cada uno de los elementos que hayamos introducido en el panel de trabajo habrá que hacer doble click sobre cada uno de ellos. Ahora bien, si queremos conectar dos elementos de ese panel de trabajo nos situaríamos sobre uno de ellos y haciendo click con el botón derecho del ratón estableceremos la orden que vamos a realizar y gráficamente seleccionaremos sobre qué elemento vamos a aplicarlo.

Por ejemplo, si queremos realizar un proceso de validación cruzada sobre un conjunto de datos almacenados en un fichero *arff,* deberemos introducir esos dos elementos en el panel de trabajo, uno a través de la pestaña *DataSources* (seleccionando *Arff Loader*) y el otro a través de la pestaña *Evaluation* (seleccionando *CrossValidation*).

Una vez que hemos introducido el fichero con el que vamos a trabajar en el elemento *Arff Loader* y dejado por defecto la configuración de *CrossValidation* conectamos ambos elementos pulsando el botón derecho sobre el elemento *Arff Loader* y seleccionamos *Dataset* y seguidamente lo asignamos a *CrossValidation*. Este proceso se visualiza en la siguiente imagen:

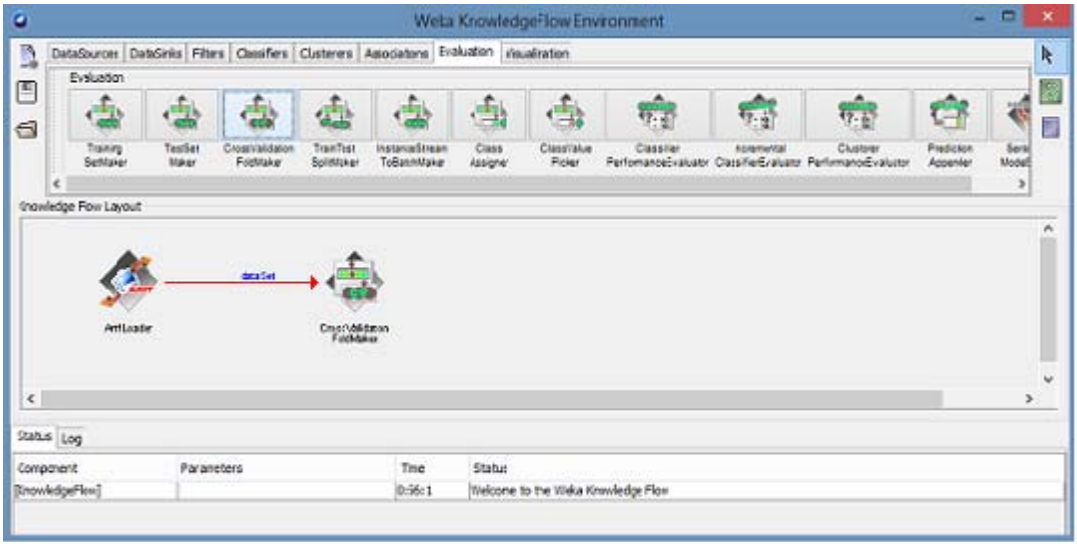


Imagen 27: Interfaz de KnowledgeFlow una vez introducidos dos elementos

Siguiendo este proceso podremos construir un proceso completo y finalmente, si queremos ejecutar el experimenter seleccionaremos la opción Start Loading en ArffLoader tal que al ejecutarse podremos ver los resultados obtenidos en cualquiera de los dos visores finales.

Finalmente y a modo indicativo, algunas de las componentes más utilizadas en las áreas de visualización y evaluación son:

a) Área Visualization:

- *DataVisualizer:* Visualiza datos en 2D.
- *AtributteSummarizer*: Histogramas, uno por atributo.
- *ModelPerformanceChart*: Curvas ROC.
- *TextViewer*: Visualiza datos o modelos en texto.
- *GraphViewer*: Visualiza modelos de árboles.

b) Área Evaluation:

- *CrossValidationFoldMaker*: Divide datasets en folds.
- *TrainTestSplitMaker*: Divide un dataset en train/test.
- *ClassAsigner*: Asigna un atributo como clase.
- *ClassValuePicker*: Elige un valor como clase positiva.
- *ClassifierPerformanceEvaluator*: Recolecta estadísticas para evaluación batch.
- *IncrementalClassifierEvaluator:* Recolecta estadísticas para evaluación incremental.
- *ClustererPerformanceEvaluator*: Recolecta estadísticas para clustering.
- *PreddictionAppender*: Añade predicciones de un clasificador a un dataset.

En conclusión, la aplicación KnowledgeFlow proporciona una alternativa a Explorer para aquellos que piensan en términos de cómo los datos fluyen a través del sistema.

4. Técnicas de Clasificación aplicadas a datos obtenidos por el Centro Andaluz de Medio Ambiente

4.1. El Centro Andaluz de Medio Ambiente

El **Centro Andaluz de Medio Ambiente** (CEAMA) es un centro mixto Junta de Andalucía y Universidad de Granada, abierto a la participación y colaboración con otras instituciones de I+D y de gestión preferentemente andaluzas. Se encuentra ubicado en la ciudad de Granada y forma parte de la estructura de centros de investigación desarrollada por el Plan Andaluz de Investigación.

Este organismo nació en 1999 con la vocación de facilitar el desarrollo de líneas de investigación que, por su singularidad temática, organizativa, instrumental o carácter multidisciplinar, no puedan realizarse eficazmente en otros centros.
Desde Julio de 2011 el CEAMA ha pasado a integrarse dentro del Instituto Interuniversitario de Investigación del Sistema Tierra en Andalucía (instituto coordinado por la Universidad de Granada).

Respecto a su estructura, el CEAMA es una organización abierta, lo que permite la incorporación de grupos de investigación que de calidad debidamente acreditada, cumplan con las características anteriores y estén abiertos al trabajo en colaboración.

En 2013 los grupos de investigación permanentes son Física de la Atmósfera, Mineralogía y Geoquímica de los Ambientes Sedimentario y Metamórfico, Análisis de Cuencas, Dinámica de Fluidos Ambientales. Sección Marina y Ecología Terrestre. En esa misma fecha se desarrollan los siguientes grupos de colaboración: Análisis Estadístico de Datos Multivariantes y Procesos Estocásticos, Cálculo Estocástico e INVESPYME.

Además, el CEAMA apoya a la Investigación I+D+i a través de sus instalaciones como son el túnel de viento de capa límite, el lidar (láser), el canal ola-corriente, All-Sky Imager, etc.

Finalmente, para obtener más información sobre los grupos de trabajo, las líneas de investigación seguidas, cursos de formación y demás novedades asociadas a los proyectos desarrollados en este centro se puede visitar su web (http://www.ceama.es/)

En conclusión, el CEAMA es un centro de investigación en temas ambientales en sus procesos fundamentales y aplicados, así como de los medios y técnicas que facilitan el uso sostenible de los recursos naturales y la mejora de la calidad de vida.

4.2. Clasificadores

Aprender cómo clasificar objetos a una de las categorías o clases previamente establecidas, es una característica de la inteligencia de máximo interés para investigadores, dado que la habilidad de realizar una clasificación y de aprender a clasificar, otorga el poder de tomar decisiones.

Definición: *Sea E un conjunto de datos, el objetivo de la clasificación es aprender una función,* L: X→Y*, denominada clasificador, que represente la correspondencia existente en los ejemplos entre los vectores de entrada y el valor de salida correspondiente.*

Y es nominal, es decir, puede tomar un conjunto de valores y_1, y_2,…, y_K denominados clases o etiquetas. La función aprendida será capaz de determinar la clase para cada nuevo ejemplo sin etiquetar. El éxito de un algoritmo de aprendizaje para clasificación depende en gran medida de la calidad de los datos que se le proporcionan.

La aplicación de un algoritmo de aprendizaje clasificador tiene como objetivo extraer conocimiento de un conjunto de datos y modelar dicho conocimiento para su posterior aplicación en la toma de decisiones. Existen distintas formas de representar el modelo generado. Entre las estructuras más utilizadas están la representación proposicional, los árboles de decisión, las reglas y listas de decisión, reglas con excepciones, reglas jerárquicas de decisión, reglas difusas y probabilidades, y redes bayesianas.

4.2.1. Evaluación del rendimiento de un clasificador

Evaluar el comportamiento de los algoritmos de aprendizaje es un aspecto fundamental; no sólo es importante para comparar algoritmos entre sí, sino que en muchos casos forma parte del propio algoritmo de aprendizaje. La forma más habitual de medir la eficiencia de un clasificador es la *precisión predictiva (accuracy)*. Cada vez que se presenta un nuevo caso a un clasificador, este debe tomar una decisión sobre la etiqueta que se le va a asignar. Considerando un error como una clasificación incorrecta de un ejemplo, se puede calcular fácilmente la tasa de error, o su complementaria, la tasa de acierto.

Definición: *Se denomina precisión de un clasificador al resultado de dividir el número de clasificaciones correctas por el número total de muestras examinadas*.

La precisión es una buena estimación de cómo se va a comportar el modelo para datos desconocidos similares a los de prueba. Sin embargo, si se calcula la precisión sobre el propio conjunto de datos utilizado para generar el modelo, se obtiene con frecuencia una precisión mayor a la real, es decir, serán estimaciones muy optimistas por utilizar los mismos ejemplos en la inducción del algoritmo y en su comprobación. La idea básica es estimar el modelo con una porción de los datos y luego comprobar su validez con el resto de los datos. Esta separación es necesaria para garantizar la independencia de la medida de precisión resultante; de no ser así, la precisión del modelo será sobreestimada.

Para tener seguridad de que las predicciones sean robustas y precisas, se consideran dos etapas en el proceso de construcción de un modelo: entrenamiento y prueba, partiendo los datos en dos conjuntos, uno de entrenamiento y otro de test.

4.3. Clasificación con WEKA

Para realizar una clasificación, será necesario elegir un clasificador y configurarlo en función de las necesidades de la implementación. WEKA posee cuatro tipos de modo de prueba:

- **Use training set**: Esta opción evalúa el clasificador sobre el mismo conjunto sobre el que se construye el modelo predictivo para determinar el error, que en este caso se denomina "error de resustitución". Por tanto, esta opción puede proporcionar una estimación demasiado optimista del comportamiento del clasificador al evaluarlo sobre el mismo conjunto sobre el que se hizo el modelo.

- **Supplied test set**: Evaluación sobre un conjunto independiente. Esta opción permite cargar un conjunto nuevo de datos. Sobre cada dato se realizará una predicción de clase para contar los errores.

- **Cross-validation**: Evaluación con validación cruzada. Se realizará una validación cruzada estratificada del número de particiones dado (*Folds*). Una validación-cruzada es estratificada cuando cada una de las partes conserva las propiedades de la muestra original (porcentaje de elementos de cada clase)

- **Percentage split** : Esta opción divide los datos en dos grupos de acuerdo con el porcentaje indicado (%). El valor indicado es el porcentaje de instancias para construir el modelo, que a continuación es evaluado sobre las que se han dejado aparte. Cuando el número de instancias es suficientemente elevado, esta opción es suficiente para estimar con precisión las prestaciones del clasificador en el dominio.

En la siguiente figura se puede ver un ejemplo de clasificación haciendo uso de *Cross-validation.*

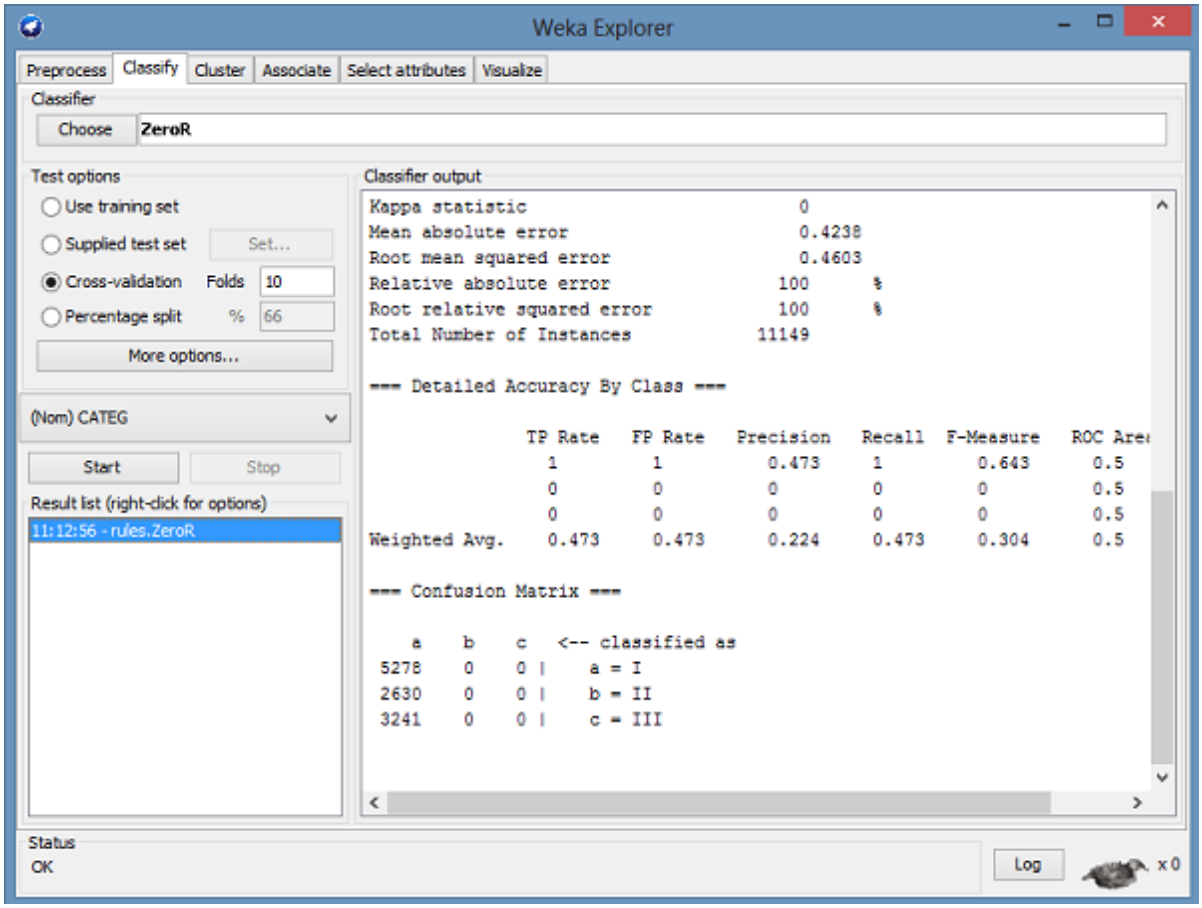


Imagen 28: Resultado de aplicar validación cruzada

Selección de clasificadores

El problema de clasificación siempre se realiza sobre un atributo simbólico, en el caso de utilizar un atributo numérico se precisa, por tanto, discretizarlo antes en intervalos que representarán los valores de clase. Existen ocho familias de clasificadores, pero los más utilizados son cuatro: los bayesianos, los metaclasificadores, las reglas y los árboles de decisión. A continuación se explicará cada uno de estos clasificadores y se pondrá un ejemplo para facilitar su comprensión.

- **Bayesianos:** La gran diferencia con otros métodos, es que cuantitativamente da una medida probabilística de la importancia de esas variables en el problema. Debe tenerse en cuenta que entre los atributos del conjunto de entrenamiento no pueden existir correlaciones, puesto que invalidaría el resultado.

    - *Naïve Bayes*: Parte de la hipótesis de que todos los atributos son independientes entre sí, conocido el valor de la variable clase. El algoritmo representa una distribución de una mezcla de componentes, donde cada componente dentro de todas las variables se asumen independientes. Esta hipótesis de independencia da lugar a un modelo de un único nodo raíz, correspondiente a la clase, y en el que todos los atributos son nodos hoja que tienen como único origen a la variable clase.

- **Metaclasificadores:** En esta familia, WEKA incluye todos aquellos clasificadores complejos, es decir, aquellos que se obtienen mediante composición de clasificadores simples o que incluyen algún preprocesamiento de los datos.

    - *Stacking*: Se basa en la combinación de modelos, construyendo un conjunto con los generados por diferentes algoritmos de aprendizaje. Como cada uno de los modelos se aprende con un mecanismo de aprendizaje diferente, se logra que los modelos del conjunto sean distintos**.**

- **Reglas**: Existen diversos métodos para generar reglas de clasificación en los conjuntos de entrenamiento.

    - *OneR*: Este es uno de los clasificadores más sencillos y rápidos, aunque en ocasiones sus resultados son sorprendentemente buenos en comparación con algoritmos mucho más complejos. Genera una regla por cada atributo y escoge la del menor error. Si hay atributos numéricos, busca los umbrales para hacer reglas con mejor tasa de aciertos.

- **Árboles de decisión:** Los árboles son una manera práctica para visualizar la clasificación de un conjunto de datos.

    - *Algoritmo J48:* Es una implementación del algoritmo C4.5, uno de los algoritmos de minería de datos que más se ha utilizado en multitud de aplicaciones. Uno de los parámetros más importantes de este algoritmo es el factor de confianza para la poda (*confidence level*). Una explicación simplificada es la siguiente: para cada operación de poda, define la probabilidad de error que se permite a la hipótesis de que el empeoramiento debido a esta operación es significativo. Cuanto más baja se haga esa probabilidad, se exigirá que la diferencia en los errores de predicción antes y después de podar sea más significativa para no podar. El valor por defecto de este factor es del 25%, y conforme va bajando se permiten más operaciones de poda y por tanto llegar a árboles cada vez más pequeños.

4.4. Datos utilizados

Los datos con los que hemos realizado este trabajo proceden del Grupo de Investigación de Física de la Atmósfera del CEAMA. Este grupo tiene entre sus objetivos:

- Medir y analizar la radiación atmosférica.
- Realizar estudios de teledetección aplicada a la caracterización de las partículas atmosféricas en suspensión y las nubes.

- Caracterización de la calidad del aire, así como el estudio y medida de los intercambios de dióxido de carbono y metano en ecosistemas terrestres.

El fichero de datos contiene 11.149 observaciones relativas a niveles de radiación y grado de nubosidad en el cielo obtenidas a través de la estación del CEAMA a diferentes horas del día durante los 12 meses del año 2011. La descripción de cada una de las variables recogidas se muestra a continuación:

Variable	Descripción
YEAR	Año
MONTH	Mes
DAY	Día
DAY_J	Día juliano
HOUR	Hora
MIN	Minuto
SZA	Ángulo Cenital Solar
AZIMU	Ángulo Azimutal
TOA	Radiación en TOA (cima de la atmósfera)
UVER	Radiación Ultravioleta eritemática
GLO	Radiación Global
DIF	Radiación Difusa
OKTAS	Oktas (medida de la nubosidad)
POR_1	Porcentaje de cielo cubierto en el octante 1
POR_2	Porcentaje de cielo cubierto en el octante 2
POR_3	Porcentaje de cielo cubierto en el octante 3
POR_4	Porcentaje de cielo cubierto en el octante 4
POR_5	Porcentaje de cielo cubierto en el octante 5
POR_6	Porcentaje de cielo cubierto en el octante 6
POR_7	Porcentaje de cielo cubierto en el octante 7
POR_8	Porcentaje de cielo cubierto en el octante 8
AOD_870	Aerosol Optical Depth 870 (profundidad óptica del aerosol)
AOD_675	Aerosol Optical Depth 675
AOD_440	Aerosol Optical Depth 440
AOD_380	Aerosol Optical Depth 380
ALPHA	Ángulo de elevación solar

4.5. Análisis de los datos

En primer lugar, a partir del fichero de datos hemos generado una variable categórica (CATEG) a partir de la hora y el minuto, con tres categorías:

- I: desde la menor hora observada hasta las 11.59.
- II: desde las 12 horas hasta las 14:59.
- III: desde las 15 horas hasta la última hora observada.

Se han eliminado del fichero las variables referentes a días y minutos; esto es, YEAR, MONTH, DAY, DAY_J, HOUR y MIN. Todos estos cambios han quedado reflejados en el fichero *simultaneos_2011.arff*.

Finalmente, el objetivo que nos planteamos con este trabajo es aplicar los métodos de clasificación descritos anteriormente para el conjunto de datos y estudiar cuales son los factores que caracterizan cada una de las categorías o grupos horarios que hemos creado.

El proceso se ha realizado a través de la interfaz Explorer de WEKA y se ha dividido en los siguientes pasos:

Paso 1: Carga del fichero de datos simultaneos_2011.arff

Para cargar el fichero se utiliza el botón OPEN FILE de la interfaz Explorer (pestaña Preprocess). Una vez seleccionado el fichero *simultaneos_2011.arff* se mostrará la siguiente pantalla:

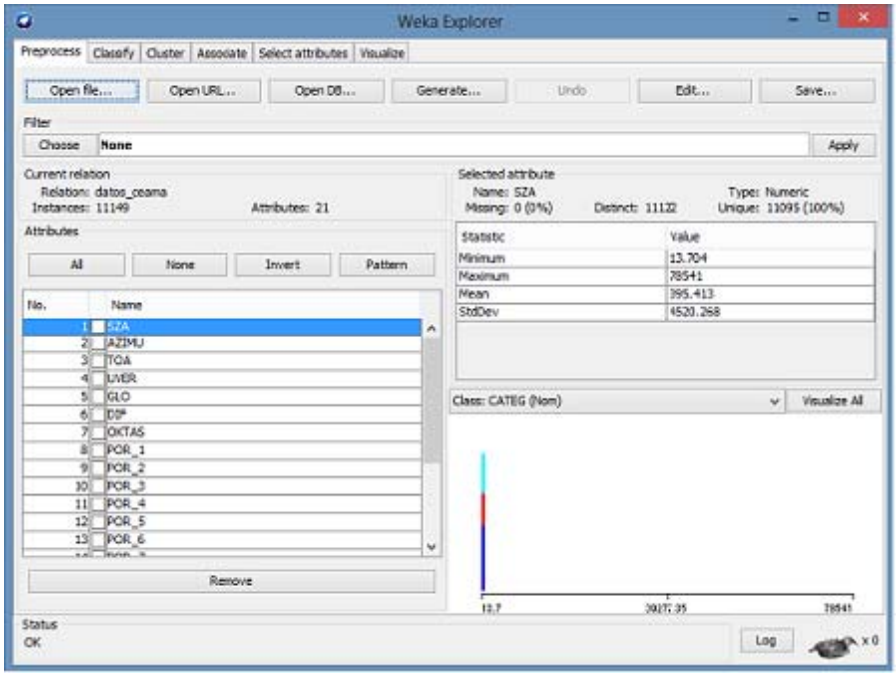


Imagen 29: Pestaña Preprocess una vez introducidos los datos

Pulsando el botón *Edit* podremos visualizar los datos del fichero con que vamos a trabajar.

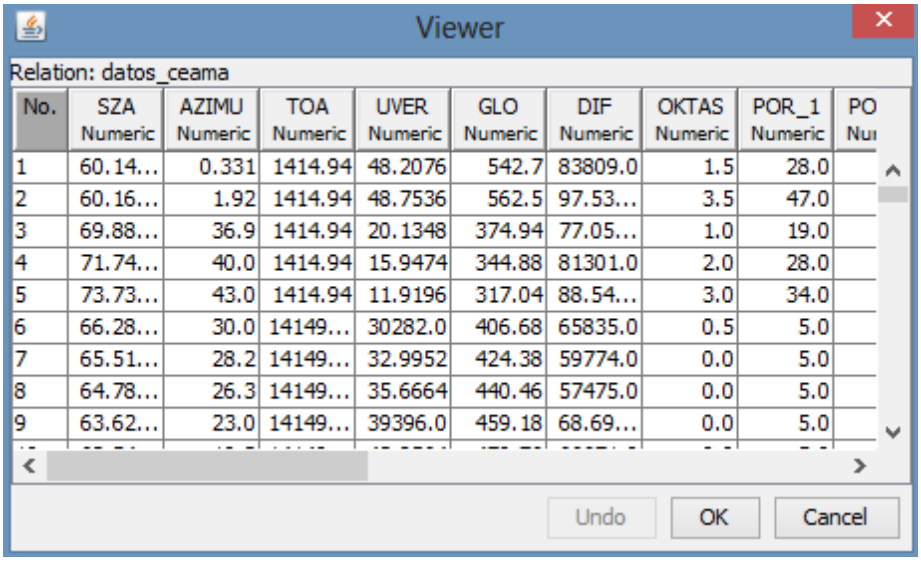


Imagen 30: Datos con los que se realizará el estudio

Paso 2: Discretización de los atributos del fichero

Al discretizar se divide el recorrido del atributo numérico en intervalos, que pueden ser de la misma amplitud o con el mismo número de observaciones (aproximadamente). En este proceso, además, se crea un atributo nominal en el que cada categoría corresponde a un intervalo.

Para realizar este proceso pulsamos el botón *Filter* y seleccionamos la opción *Discretize* dentro de los filtros no supervisados.

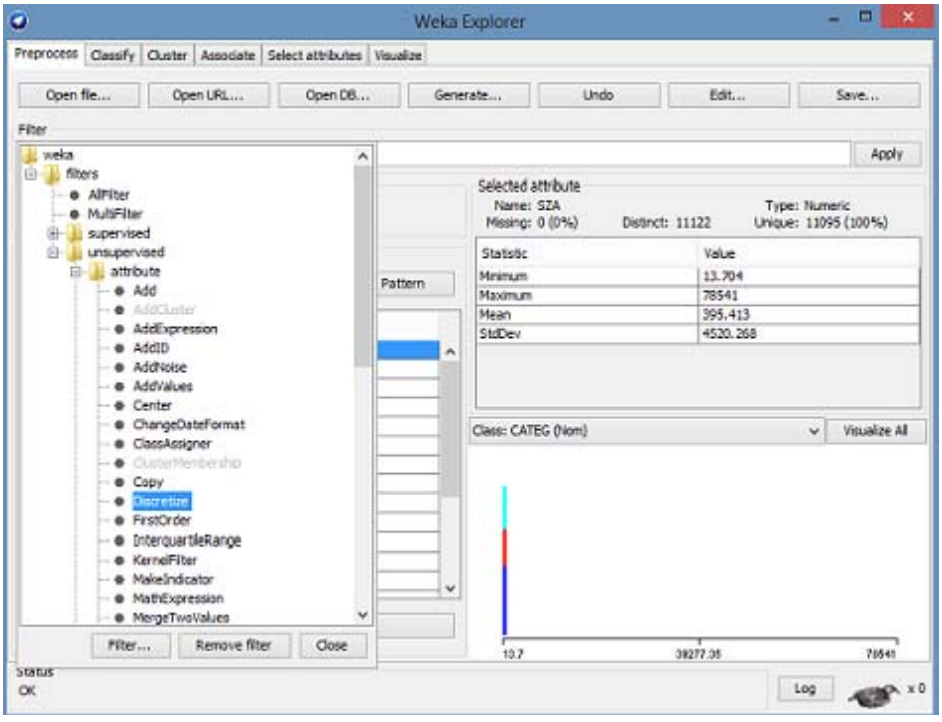


Imagen 31: Selección del filtro *Discretize*

En nuestro caso una vez seleccionado este filtro no modificaremos ninguno de sus parámetros asociados. Si lo deseamos podemos hacerlo pulsando sobre el filtro apareciendo la siguiente pantalla:

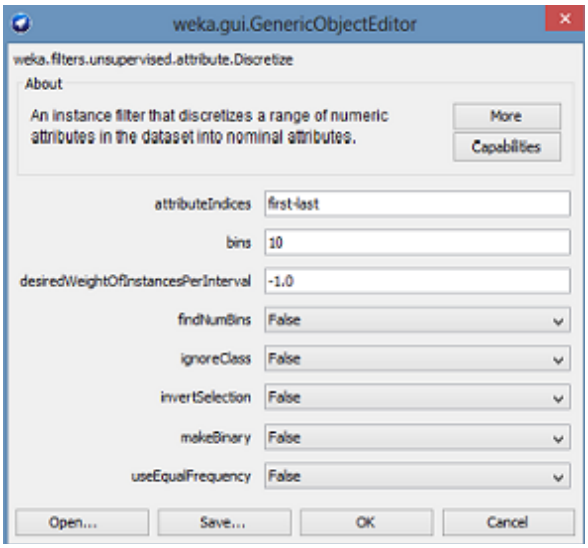


Imagen 32: Parámetros del filtro *Discretize*

donde:

- *attibuteIndices*: Índices de los atributos sobre los que actuará el filtro. Por defecto todos.
- *Bins:* Número de categorías del atributo resultante.
- *desiredWeightOfInstancePerInterval:* Número deseado de observaciones en cada intervalo.
- *FindNumBins:* Determina el número de intervalos de forma óptima.
- *InvertSelection:* Invierte la selección de atributos indicado en attributeIndices.
- *MakeBinary:* Convierte el atributo resultante en un conjunto de atributos binarios.
- *UseEqualFrequency:* Intervalos con igual frecuencia, aunque distinta amplitud.

Tal y como se ha comentado, en nuestro caso no se ha modificado ninguno de estos parámetros, por lo que una vez seleccionado el filtro y pulsado el botón *Apply* se realiza el proceso de discretización en los datos; así pues, cada observación de cada una de las variables se introducirá en su correspondiente intervalo.

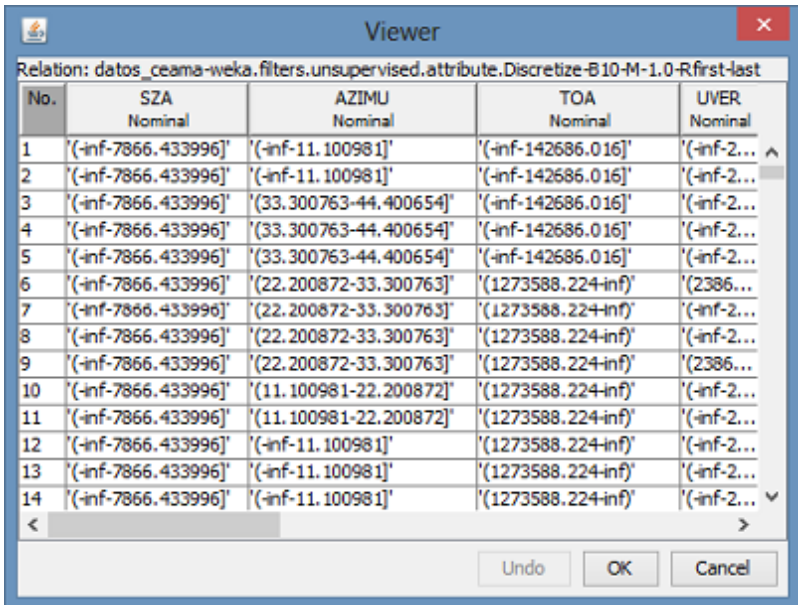


Imagen 33: Datos tras el proceso de discretización

Paso 3: Aplicación de métodos de clasificación

A) Método de clasificación Naïve Bayes

Tal y como se ha descrito anteriormente, este método de clasificación forma parte de los clasificadores Bayesianos; es decir, basados en el Teorema de Bayes. Esta técnica involucra una hipótesis de difícil cumplimiento y funciona bien con bases de datos reales; en especial, cuando se combina con procedimientos de selección de atributos para eliminar la redundancia.

En nuestro caso, para aplicar este método habrá que dirigirse a la pestaña *Classify* y tras pulsar el botón *Choose*, seleccionarlo tal y como se muestra en la Imagen 34.

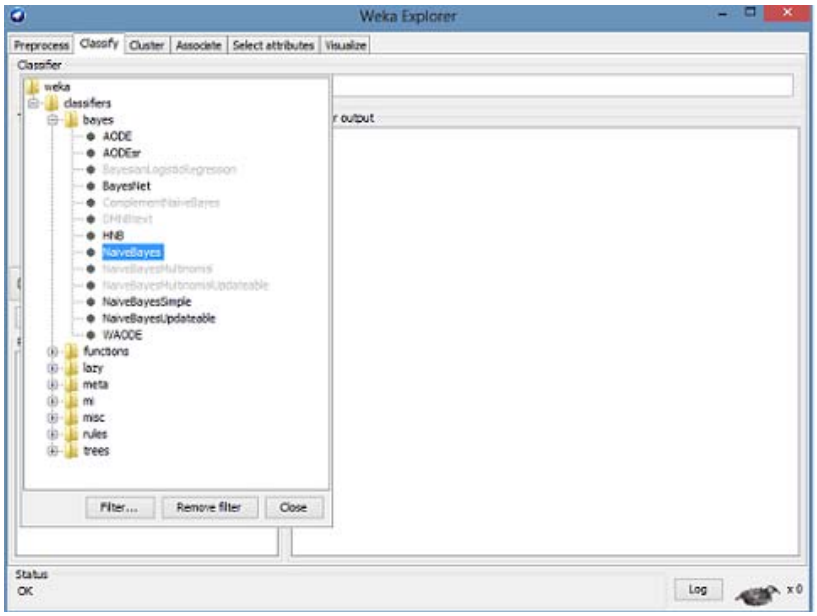


Imagen 34: Selección del clasificador NaiveBayes

Una vez elegido realizaremos este proceso para tres de los cuatro métodos de prueba descritos anteriormente. No lo haremos para caso *Supplied test set* puesto que no poseemos un conjunto independiente sobre el que evaluar los resultados. Así pues tenemos:

A1) Usando un conjunto de entrenamiento. Al seleccionar la opción *Use training set* y pulsar el botón *Start* se muestra la siguiente información:

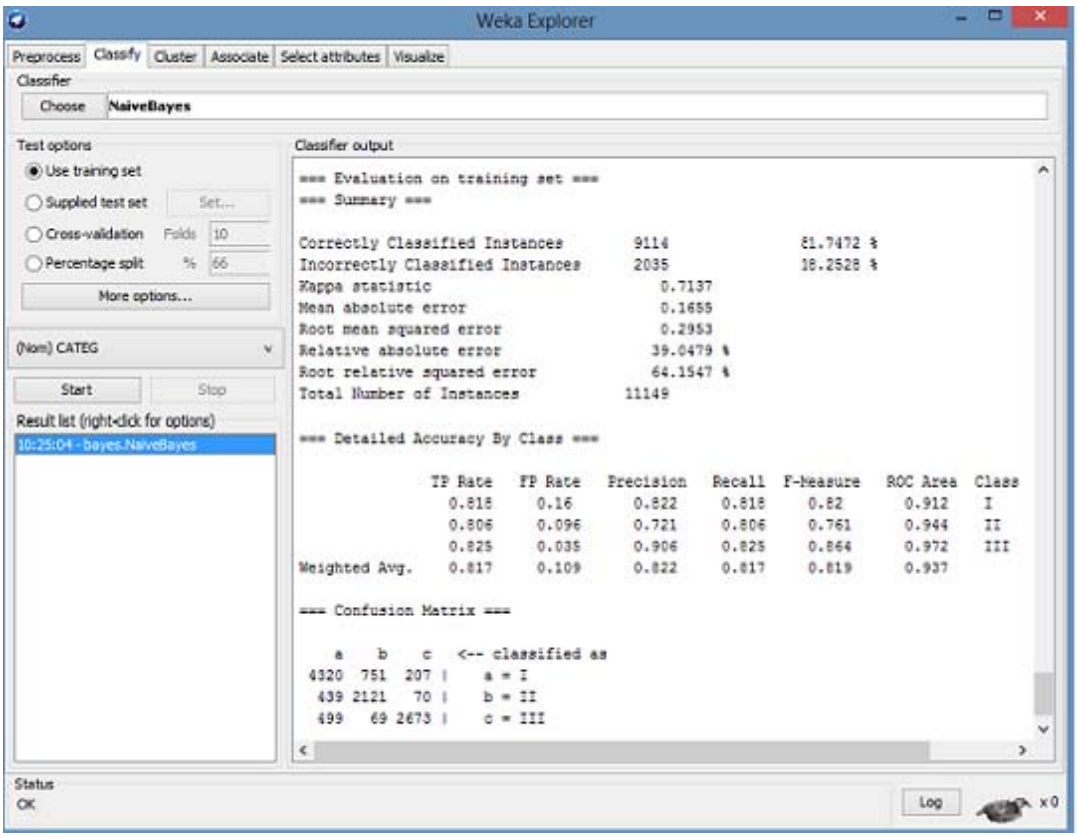


Imagen 35: Salida de aplicar NaiveBayes y un conjunto de entrenamiento

Seguidamente vamos a analizar la salida resultante de aplicar el clasificador. De esa forma vamos a definir los términos que allí aparecen y que son útiles para comparar los distintos métodos de clasificación que vamos a aplicar.

En la Imagen 35 se muestra que el 81,7472% de los casos se han clasificado correctamente mientras que el 18,2528% lo han hecho de forma incorrecta.

Seguidamente se muestra el valor del índice Kappa (0,7137). Esté índice es una medida de concordancia entre las categorías pronosticadas por el clasificador y las categorías observadas, que tiene en cuenta las posibles concordancias debidas al azar. Donde:

- Si el valor es 1 : Concordancia perfecta.
- Si el valor es 0 : Concordancia debida al azar.
- Si el valor es negativo: Concordancia menor que la que cabría esperar por azar.

Por lo tanto, en nuestro caso, tenemos un alto grado de concordancia.

Posteriormente aparecen ciertas medidas asociadas al error de la clasificación. Estos coeficientes se calculan a partir de unos valores d_i que se obtienen de la siguiente forma para cada instancia:

- ✓ Se construye un vector binario que tiene un uno en la posición de la clase a la que pertenece la instancia y ceros en las demás.
- ✓ Se determina el vector de probabilidades de asignación a las distintas clases que proporciona el clasificador.
- ✓ Se realiza la diferencia entre el par de vectores asociados.
- ✓ Las componentes de los vectores diferencia proporcionan los valores d_i.

Para visualizar los valores d_i bastaría con marcar la opción *Output predictions* si pulsamos el botón *More options* y seguidamente *Start*.

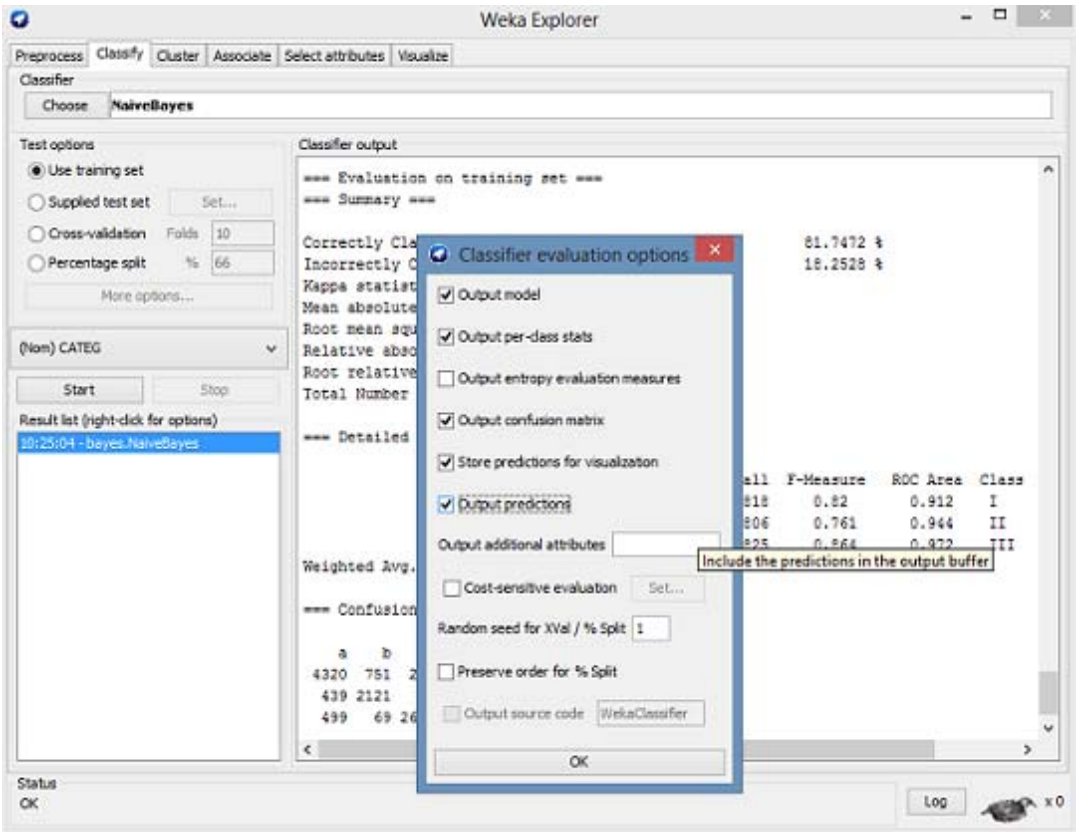


Imagen 36: Selección de la opción *Output predictions*

Los indicadores asociados al error de la clasificación son:

-Mean absolute error:

$$\frac{1}{N}\sum_i |d_i| = 0.1655.$$

- Root mean squared error:

$$\sqrt{\frac{1}{N}\sum_i d_i^2} = 0.2953.$$

- Relative absolute error (%):

$$\frac{Mean\ absolute\ error}{Mean\ absolute\ error\ (ZeroR)} \times 100 = 39.0479$$

- Root relative squared error (%):

$$\frac{Root\ mean\ squared\ error}{Root\ mean\ squared\ error\ (ZeroR)} \times 100 = 64.1547$$

Seguidamente aparecen el número total de casos (11.149) y un cuadro donde se muestran una serie de indicadores relativos a la precisión de la clasificación. Estos coeficientes se basan en los resultados de la matriz de confusión que aparece al final de la salida. En este caso, la matriz de confusión muestra la siguiente información:

```
== Confusion Matrix ===
   a    b    c  <-- classified as
4320  751  207 |  a = I
 439 2121   70 |  b = II
 499   69 2673 |  c = III
```

Es decir, de los 5.278 casos del grupo I, 4.320 se han clasificado correctamente, 751 se han clasificado dentro del grupo II y 207 se han clasificado dentro del grupo III. A modo de resumen podemos clasificar estos valores como:

- ✓ **Verdaderos positivos (TP)**: Instancias correctamente reconocidas por el sistema. Corresponden a los valores de la diagonal (4.320, 2.121 y 2.673).

- ✓ **Verdaderos negativos (TN):** Instancias que son negativas y correctamente reconocidas como tales. Si consideramos únicamente el estudio para una clase, por ejemplo para la clase I, entonces los verdaderos negativos serían 2.121 y 2.673.

- ✓ **Falsos positivos (FP):** Instancias que son negativas pero el sistema dice que no lo son. Por ejemplo, tenemos 439 casos que pertenecen al grupo II y que han sido clasificadas como I.

- ✓ **Falsos negativos (TN):** Instancias que son positivas y que el sistema dice que no lo son. Por ejemplo existen 751 casos del grupo I que se han clasificado dentro del grupo II.

A partir de estos valores se calculan los indicadores de precisión para cada clase, que se definen como:

- **Tasa de verdaderos positivos**:

$$\text{TP rate} = \frac{TP}{TP + FN}$$

- **Tasa de falsos positivo**s:

$$\text{FPrate} = \frac{FP}{FP + TN}$$

- **Medida de precisión**:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**:

$$\text{Recall(X)} = \frac{\text{Nº de instancias de la clase X clasificadas correctamente}}{\text{Nº de instancias perteneciente a X}}$$

y es equivalente a:

$$\text{Recall} = \text{TP rate}$$

- **Medida F**:

$$\text{F-measure} = \frac{2TP}{2TP + FP + FN}$$

El resultado de estos indicadores se muestra en la salida del clasificador, obteniendo:

== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.818	0.16	0.822	0.818	0.82	0.912	I
	0.806	0.096	0.721	0.806	0.761	0.944	II
	0.825	0.035	0.906	0.825	0.864	0.972	III
Weighted Avg.	0.817	0.109	0.822	0.817	0.819	0.937	

Mejora de precisión del clasificador

Hay que hacer que notar que un ejemplo de método de aprendizaje que reduce su calidad ante la presencia de atributos no relevantes es el método Naive Bayes. Tal y

como hemos visto, al utilizar este método mediante un conjunto de entrenamiento hemos obtenido una precisión del 81,7472%.

Seguidamente vamos a comprobar si los atributos no relevantes están afectando a la calidad del método, por lo que vamos a efectuar un filtrado de atributos. Para ello vamos a la sección *Select Attributes* donde encontramos dos familias de técnicas para realizar este proceso:

- **Filtros,** donde se seleccionan y evalúan los atributos en forma independiente del algoritmo de aprendizaje.
- **Wrappers** (envoltorios), los cuales usan el desempeño de algún clasificador para determinar lo deseable de un subconjunto.

Dadas las características del problema en este caso podemos probar con una técnica *wrapper* realizando una búsqueda exhaustiva. Para ello, pulsamos *Choose* de *AttributeEvaluator* y seleccionamos el método *WrapperSubsetEval*. Para configurarlo pulsamos en la ventana de texto. Vamos a utilizar el propio *NaiveBayes* para el *wrapper*, por lo que seleccionaremos el método en *classifier*. Por otra parte, en *SearchMethod* indicamos que queremos una búsqueda exhaustiva eligiendo *ExhaustiveSearch*. Una vez configurada la herramienta, pulsamos el botón Start, mostrándose la información de la Imagen 37.

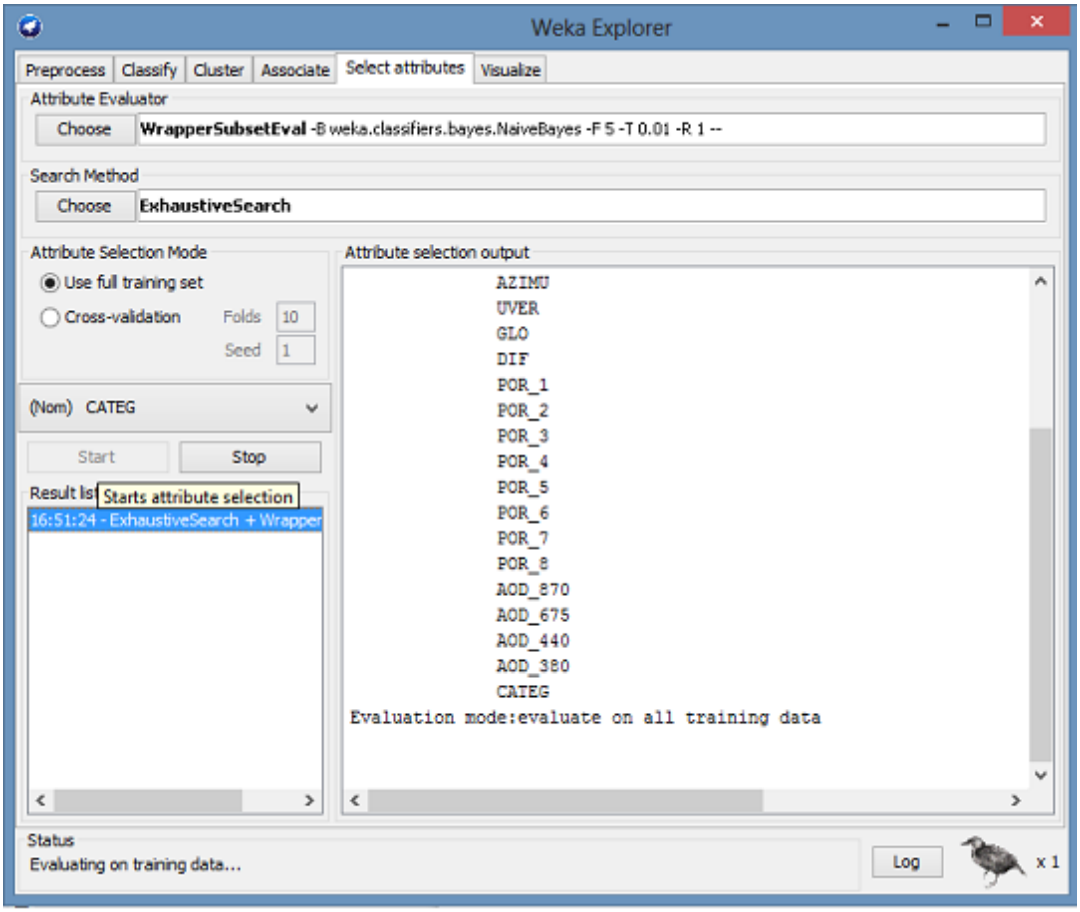


Imagen 37: Pestaña de selección de atributos

Analizando los resultados de este método de filtraje se recomienda no utilizar los atributos TOA, OKTAS y ALPHA. Para comprobar la veracidad de esta recomendación

volvemos a la pantalla *Preprocess*, y eliminamos los atributos descartados marcándolos en la parte de *Attributes* y pulsamos en *Remove*.

Finalmente, para conocer la precisión que obtiene *NaiveBayes* con este subconjunto de atributos, volvemos a la ventana *Classify* y seleccionamos el método *NaiveBayes* usando un conjunto de entrenamiento. El resultado muestra que el porcentaje de instancias correctamente clasificadas ha aumentado al 82,0522%. El resto de indicadores se pueden visualizar en la siguiente imagen:

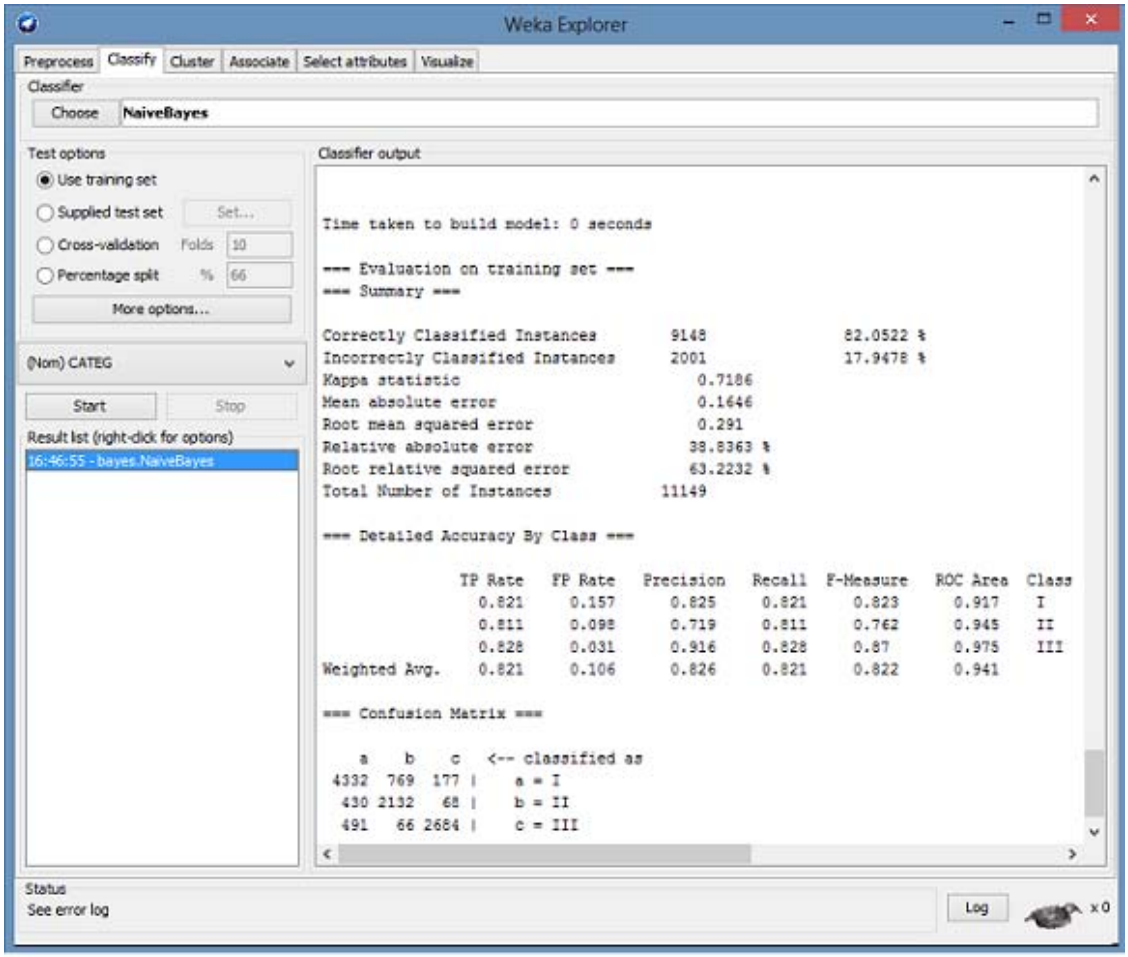


Imagen 38: Clasificador NaiveBayes con un conjunto de entrenamiento

A2) Usando validación cruzada. Supuesto que hemos eliminado aquellos atributos que disminuyen la precisión de los estimadores de igual forma que lo hemos hecho en el apartado anterior; seguidamente, seleccionamos el método Cross-validation, manteniendo el número de pliegues que aparecen por defecto (10). Pulsando el botón Start se muestra la siguiente información:

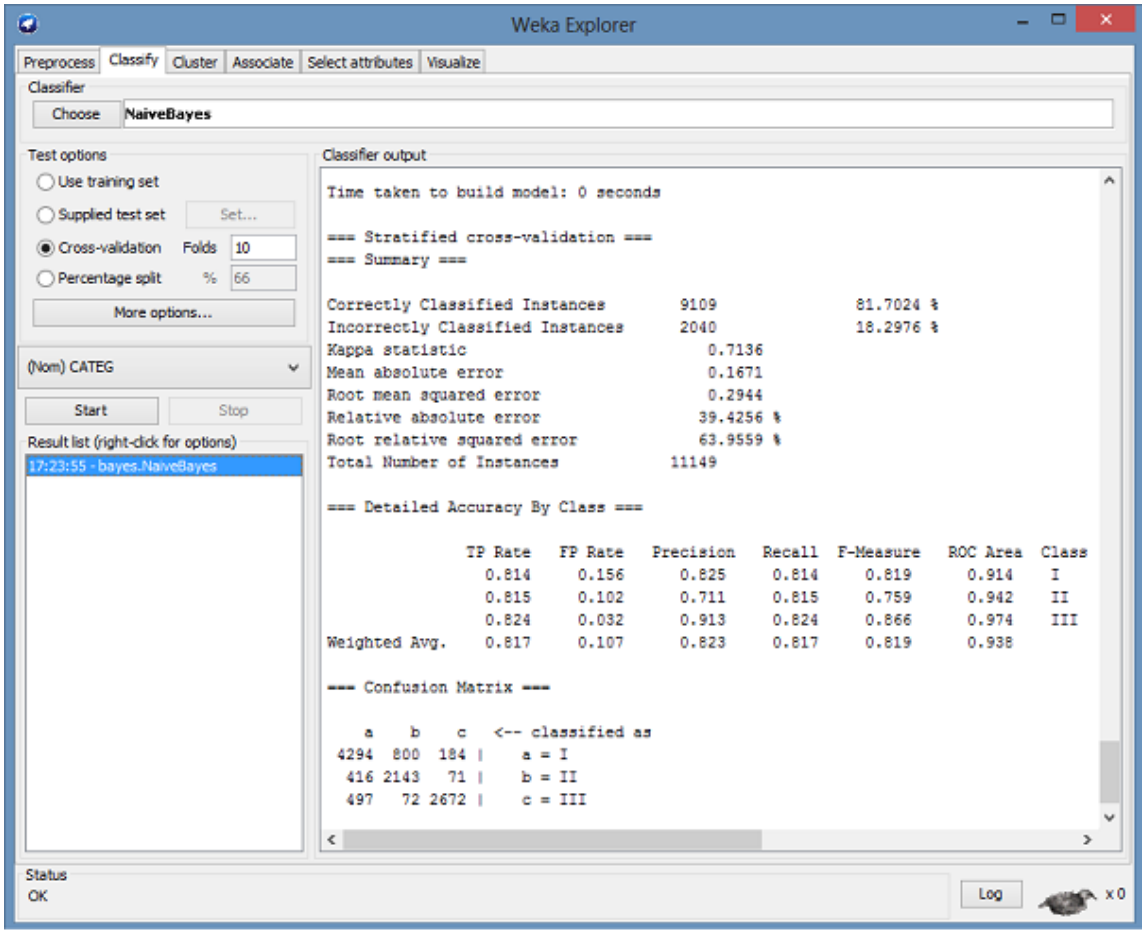


Imagen 39: Clasificador NaiveBayes y validación cruzada

En este caso, el porcentaje de instancias clasificadas como correctas ha sido del 81,7024%, mientras que el de clasificadas incorrectamente es del 18,2976%. El índice Kappa toma el valor 0,7136 por lo que existe un alto grado de concordancia.

Finalmente, basándonos en los resultados de la matriz de confusión y a modo de ejemplo tenemos que de los 3.241 casos pertenecientes al grupo III, 2.672 se han clasificado correctamente, mientras 497 se han clasificado como I y 72 como II.

A3) Dividiendo el fichero de datos. Al igual que en los apartados anteriores hemos eliminado aquellos atributos que disminuyen la precisión de los estimadores.

Seguidamente seleccionamos el método *Percentage split*. A través de este método el fichero de datos se divide en dos partes, de acuerdo al porcentaje indicado (que dejaremos en el 66%). Una se usa para construir el clasificador y la otra para evaluar su rendimiento.

Tras pulsar el botón Start aparece la información que se muestra en la siguiente imagen.

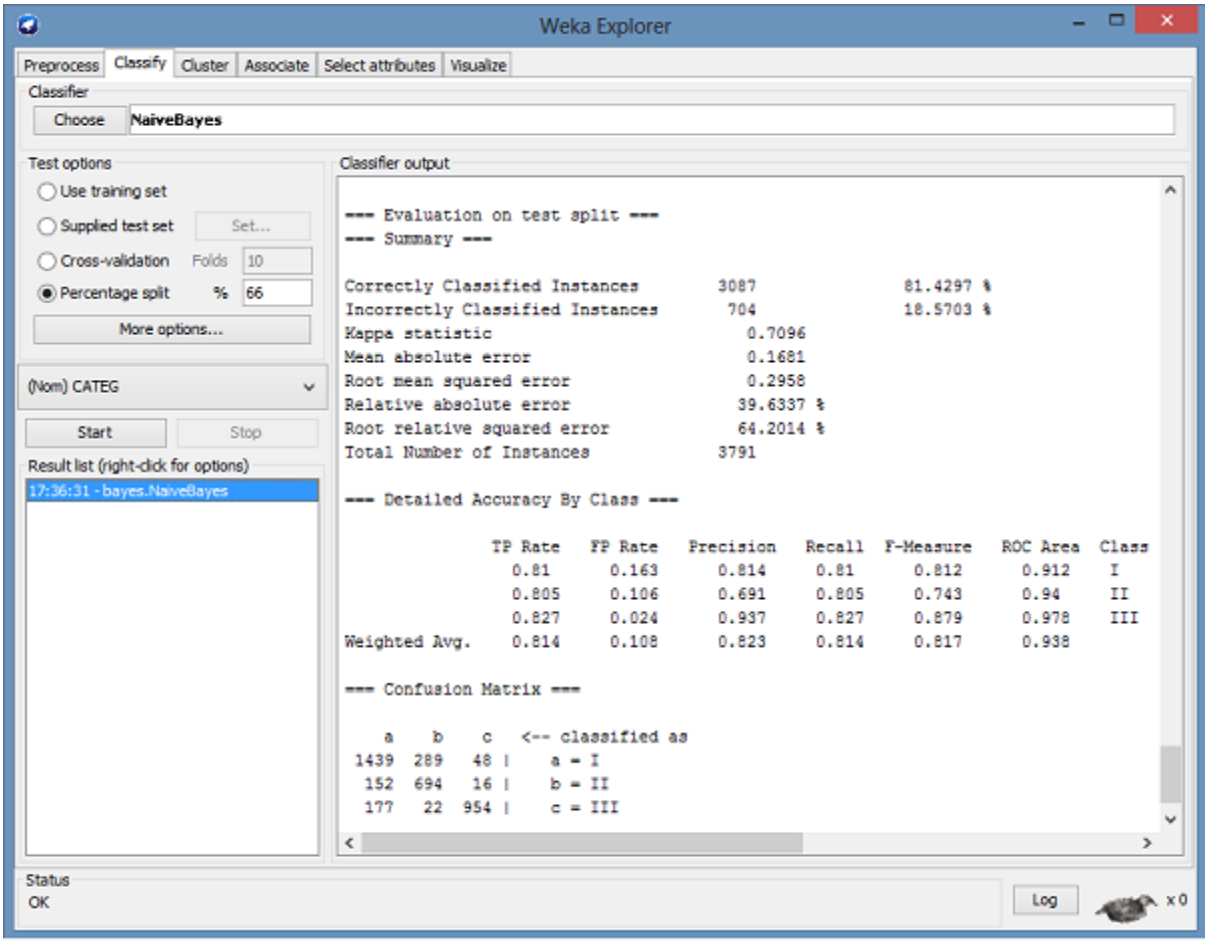


Imagen 40: Clasificador NaiveBayes y Percentage split

En este caso, el porcentaje de instancias clasificadas como correctas disminuye respecto al anterior modelo utilizado pasando al 81,4297%, mientras que el de clasificadas incorrectamente es del 18,5703%. El índice Kappa toma el valor 0,796 por lo que existe un alto grado de concordancia.

Por otro lado y a modo de ejemplo, la tasa de instancias clasificadas correctamente para la clase I es de 0,81. Finalmente, de los 862 casos pertenecientes al grupo II, 694 se han clasificado correctamente, mientras 152 se han clasificado como I y 16 como III.

B) Método de clasificación Stacking

Tal y como hemos indicado anteriormente, Stacking es un meta-clasificador, de estructura bastante sencilla, que se basa en la combinación de modelos, construyendo un conjunto con los generados por diferentes algoritmos de aprendizaje. Como cada uno de los modelos aprende a través de un mecanismo de aprendizaje diferente, se logra que los modelos del conjunto sean distintos.

Para nuestros propósitos, vamos a definir tres clasificadores base (*NaiveBayes, OneR y J48*) y utilizar el clasificador J48 como meta clasificador para el conjunto de datos, *simultaneos_2011.arff,* que hemos discretizado en el apartado anterior.

Comenzamos pulsando la pestaña *Classify* y pulsando sobre el botón *Choose* seleccionamos el clasificador *Stocking*. Una vez elegido modificamos los parámetros eligiendo los clasificadores base y el meta-clasificador definidos en el párrafo anterior.

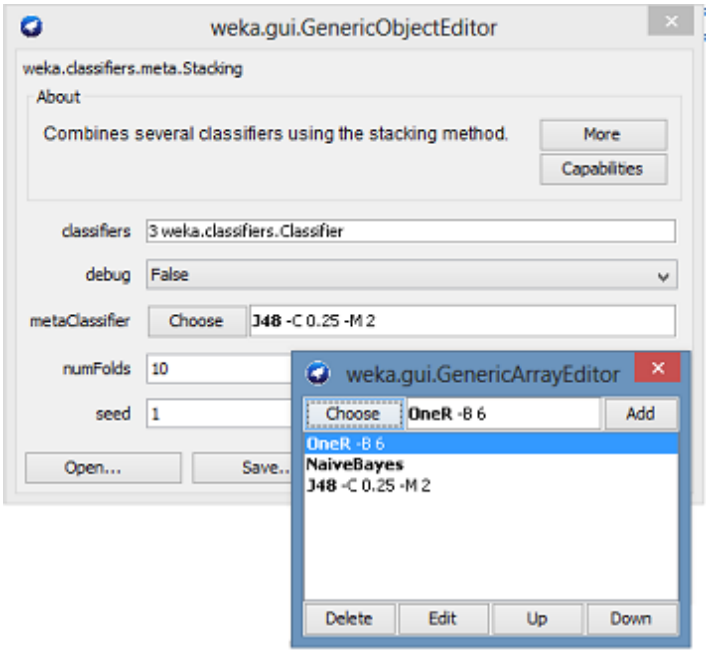


Imagen 41: Selección de clasificadores base y meta clasificador para el método Stocking

Una vez confirmados estos cambios realizaremos este proceso para los tres métodos de prueba con los que hemos realizado el anterior proceso de clasificación. Así pues tenemos:

B1) Usando un conjunto de entrenamiento. Al seleccionar la opción *Use training set* y pulsar el botón *Start* se muestran los modelos inducidos para cada clasificador individual y para el modelo aprendido por el meta clasificador. El resultado final se muestra en la Imagen 42.

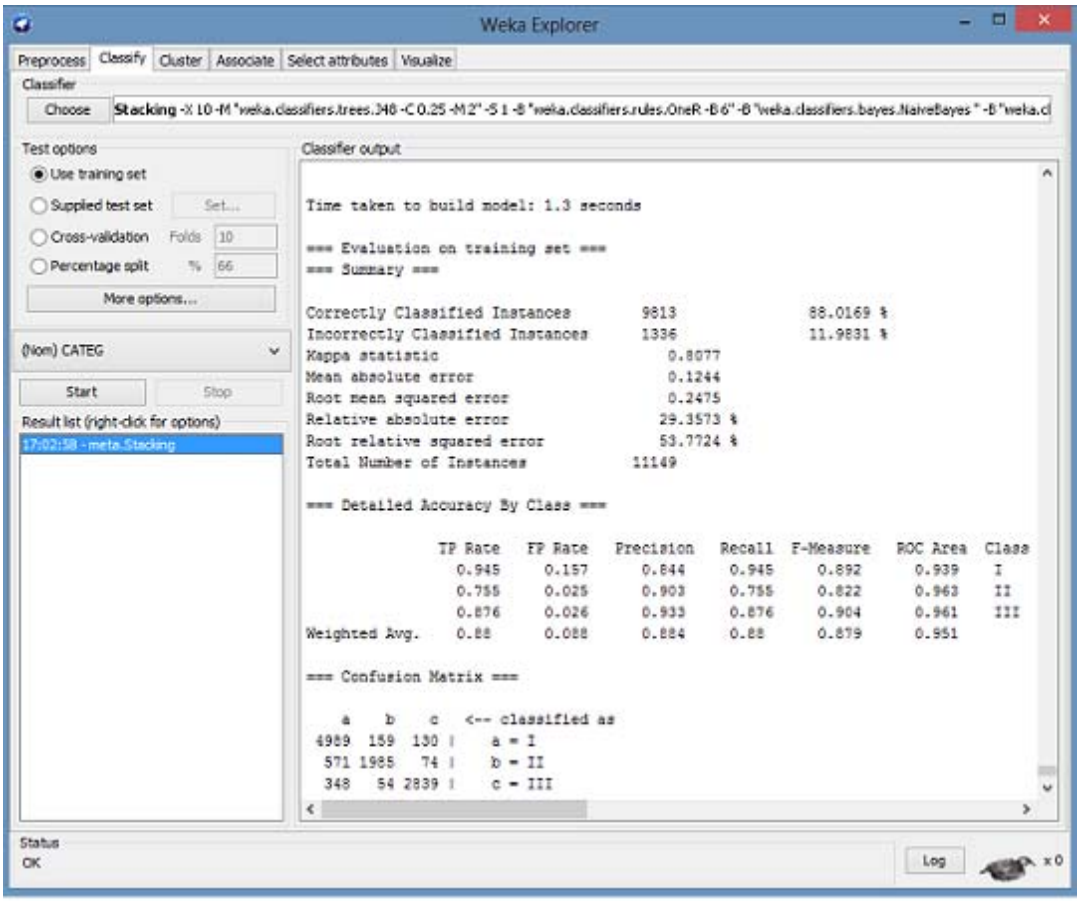


Imagen 42: Resumen de los resultados de aplicar el método Stocking y un conjunto de entrenamiento

Analizando los resultados se tiene que el porcentaje de instancias clasificadas como correctas aumenta respecto al anterior método de clasificación obteniendo un 88,0169%, mientras que el de clasificadas incorrectamente es del 11,9831%. El índice Kappa toma el valor 0,8077 por lo que existe un alto grado de concordancia.

Por otro lado y a modo de ejemplo la tasa de instancias clasificadas correctamente para la clase I es de 0,945. Finalmente, de los 2.633 casos pertenecientes al grupo II, 1.985 se han clasificado correctamente, mientras 571 se han clasificado como I y 74 como III.

B2) Usando validación cruzada. Para este apartado mantenemos la misma elección de clasificadores base y meta clasificador que hemos realizado en el punto B1.

Para aplicar este método pulsamos la opción *Cross-validation* dentro de la sección *Test Option*. Además mantendremos el número de pliegues que aparecen por defecto (10) y pulsando el botón Start se muestra la siguiente información:

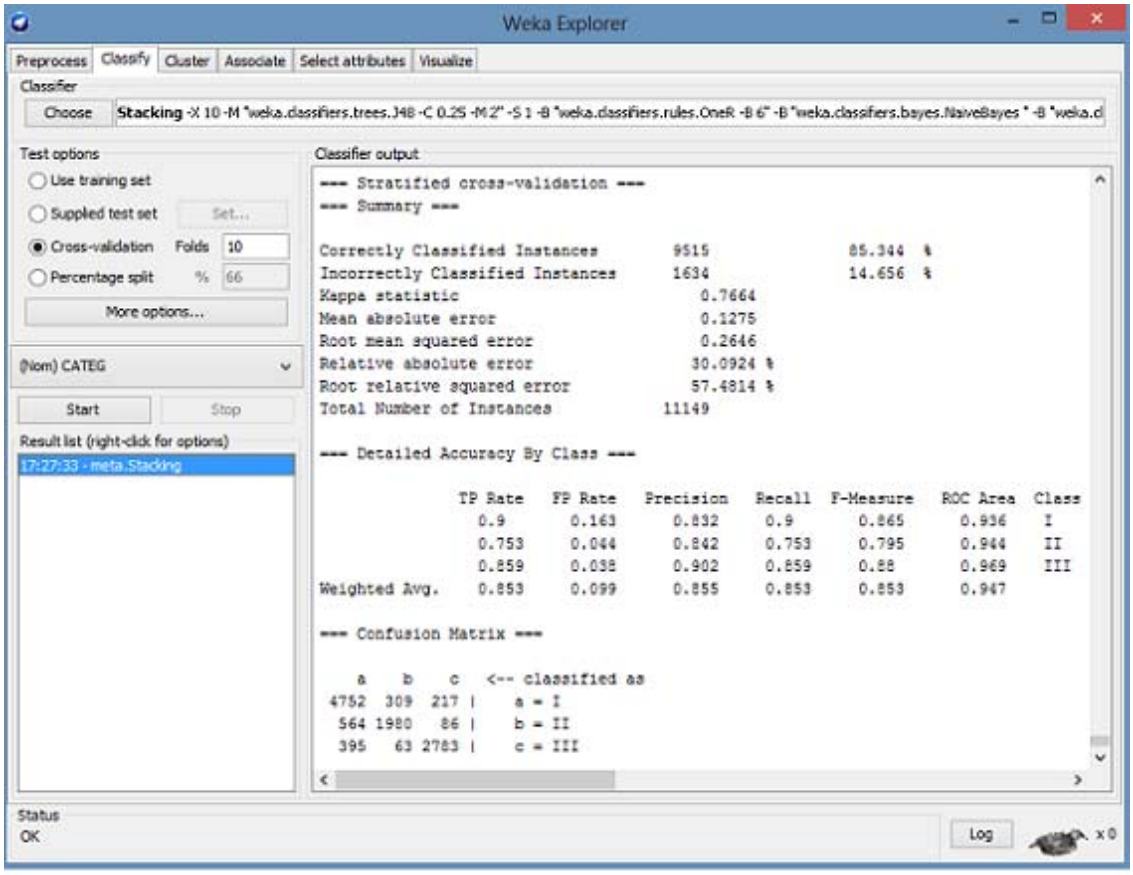


Imagen 43: Resumen de los resultados de aplicar el método Stocking y validación cruzada

A través de los resultados de la Imagen 43 se aprecia que el número de casos clasificados correctamente disminuye respecto al anterior modo de prueba, obteniendo un porcentaje del 85,344%. El 14,656% de los casos o instancias se han clasificado de forma incorrecta.

El valor del índice Kappa (0,7664) también disminuye respecto al anterior modo de prueba.

A modo de ejemplo, la matriz de confusión muestra que de los 5.278 casos del grupo I, 4.752 se han clasificado de forma correcta, 309 lo han hecho como del grupo II y 217 como del grupo III.

B3) Dividiendo el fichero de datos. Para realizar este apartado seleccionaremos el modo de prueba *Percentage split*. A través de este método, el fichero de datos se divide en dos partes, de acuerdo al porcentaje indicado (que dejaremos en el 66%). La primera se utilizará para construir el clasificador y la segunda para evaluar su rendimiento.

Tras pulsar el botón Start aparece la información que se muestra en la siguiente imagen.

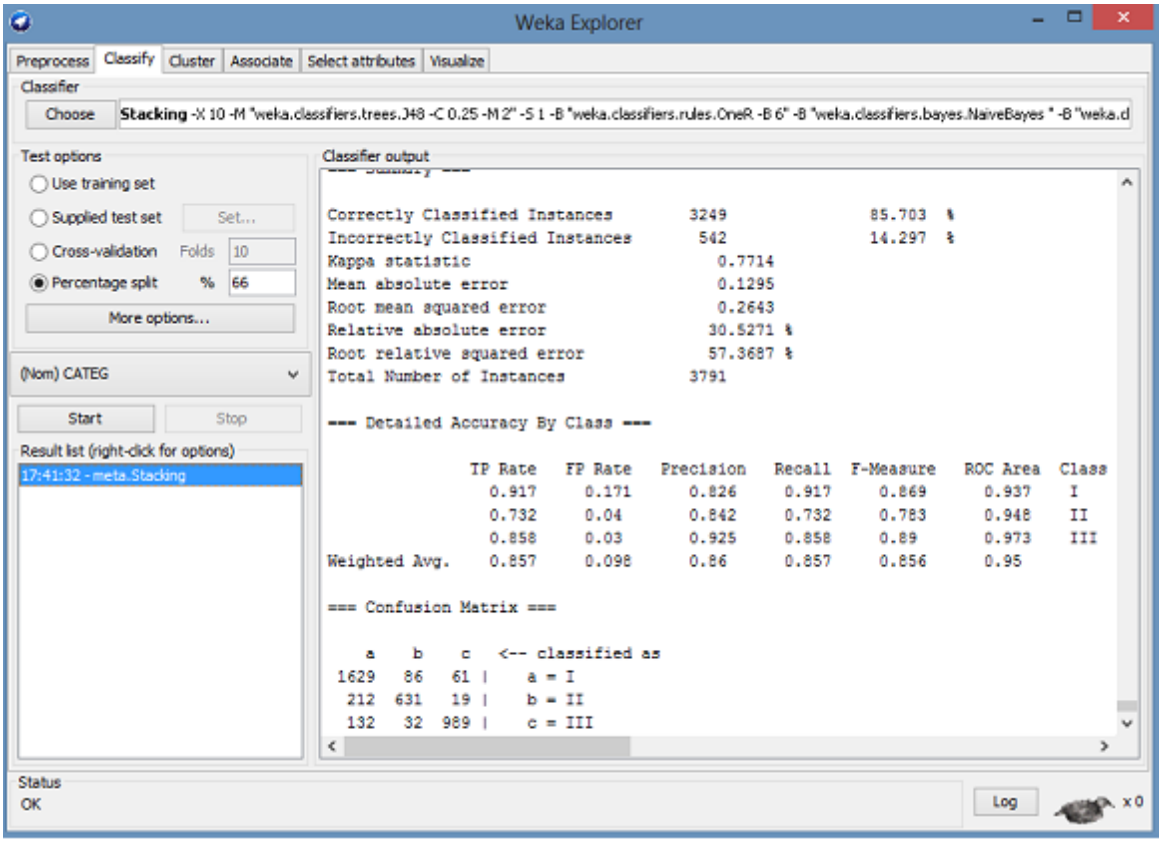


Imagen 44: Resumen de los resultados de aplicar el método Stocking con división del fichero

En este caso, el porcentaje de instancias clasificadas correctamente es de un 85,703% mientras que el 14,297% lo han sido de forma errónea, teniendo como porcentaje de error absoluto relativo 30,5271%.

El valor del índice Kappa es de 0,7714 por lo que existe alto grado de concordancia entre las categorías pronosticadas por el clasificador y las categorías observadas.

Finalmente, a través de la matriz de confusión se obtiene entre otras cosas, que de los 1.153 casos del grupo III, 989 se han clasificado correctamente, 132 se han clasificado como del grupo I y 32 lo han hecho como del grupo II.

C) Método de clasificación OneR

Este clasificador es uno de los más sencillos y rápidos. Sus resultados pueden ser muy buenos en comparación con otros algoritmos mucho más complejos y su objetivo es seleccionar el atributo que mejor explica la clase de salida.

Para seleccionar este método habrá que elegirlo dentro de la sección *rules* al pulsar sobre el botón *Choose* que se encuentra dentro del apartado *Classifier*.

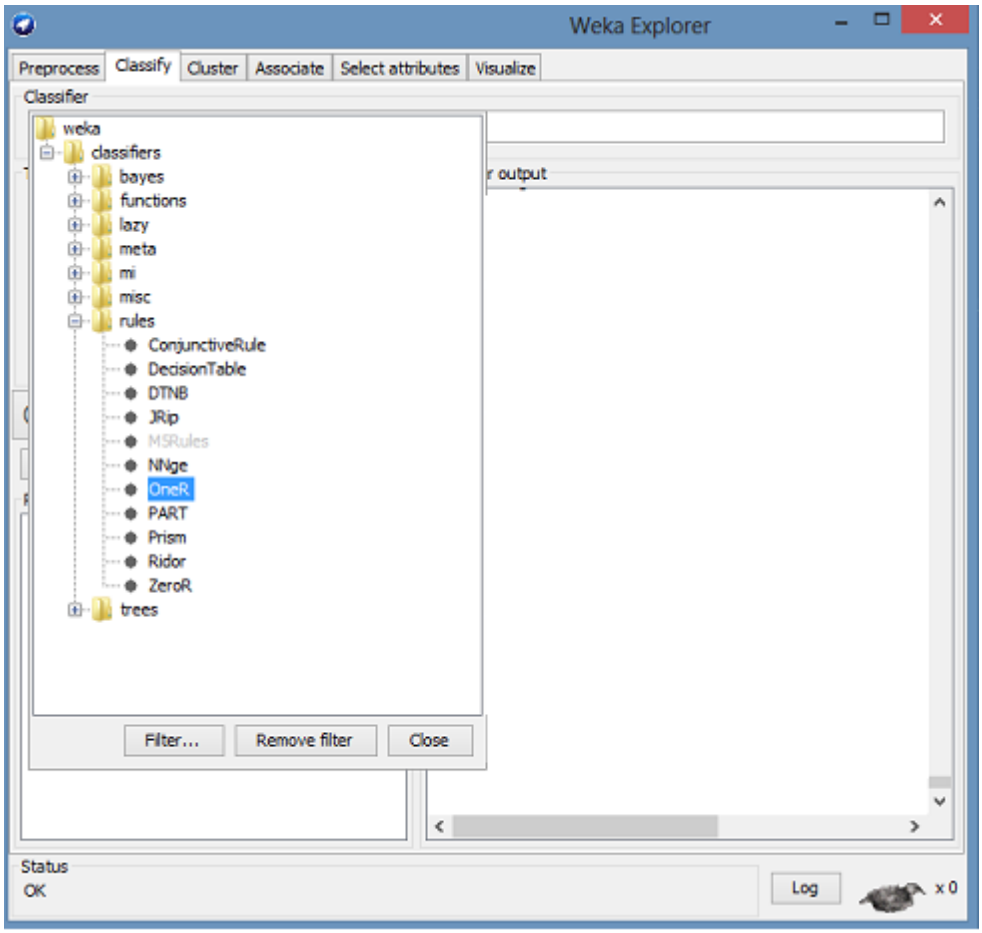


Imagen 45: Selección del método OneR

Una vez seleccionado el método de clasificación y para cada modo de prueba obtenemos los siguientes resultados:

C1) Usando un conjunto de entrenamiento. Al seleccionar la opción *Use training set* y pulsar el botón *Start* se muestra información sobre el atributo que mejor explica la clase de salida, en este caso, POR_3 e indicadores asociados la calidad de la clasificación tal y como se muestra en la Imagen 46.

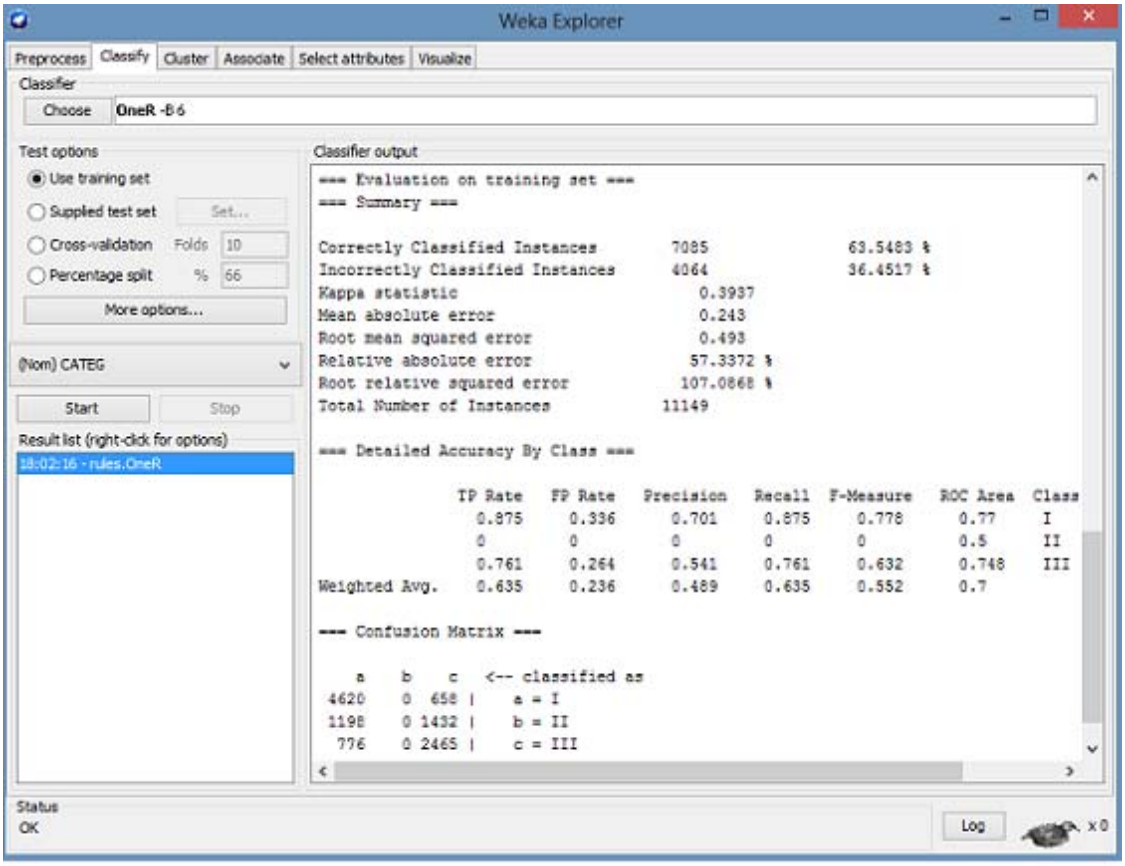


Imagen 46: Aplicación del clasificador OneR y un conjunto de entrenamiento

Analizando estos resultados, vemos que la mejor predicción posible con un solo atributo es la variable POR_3 con los siguientes umbrales:

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

POR_3:
'(-inf-9.9]' -> I
'(9.9-19.8]' -> III
'(19.8-29.7]' -> III
'(29.7-39.6]' -> III
'(39.6-49.5]' -> III
'(49.5-59.4]' -> III
'(59.4-69.3]' -> III
'(69.3-79.2]' -> III
'(79.2-89.1]' -> III
'(89.1-inf)' -> III
(7085/11149 instances correct)

Es decir, por debajo del valor 9,9 las instancias se clasificarán como I por encima de ese valor como III. No encontrando ningún valor para clasificar dentro del grupo II.

La tasa de aciertos sobre el propio conjunto de entrenamiento es del 63.5483% y respecto a la matriz de confusión tenemos que de los 2.630 casos del grupo II, 1.198 se han clasificado como I y 1.432 como III, por lo que no existe ninguno de estos casos que se haya clasificado de forma correcta.

C2) Usando validación cruzada. En este caso seleccionamos el método *Cross-validation*, manteniendo el número de pliegues que aparecen por defecto (10). Pulsando el botón *Start* se muestra la siguiente información:

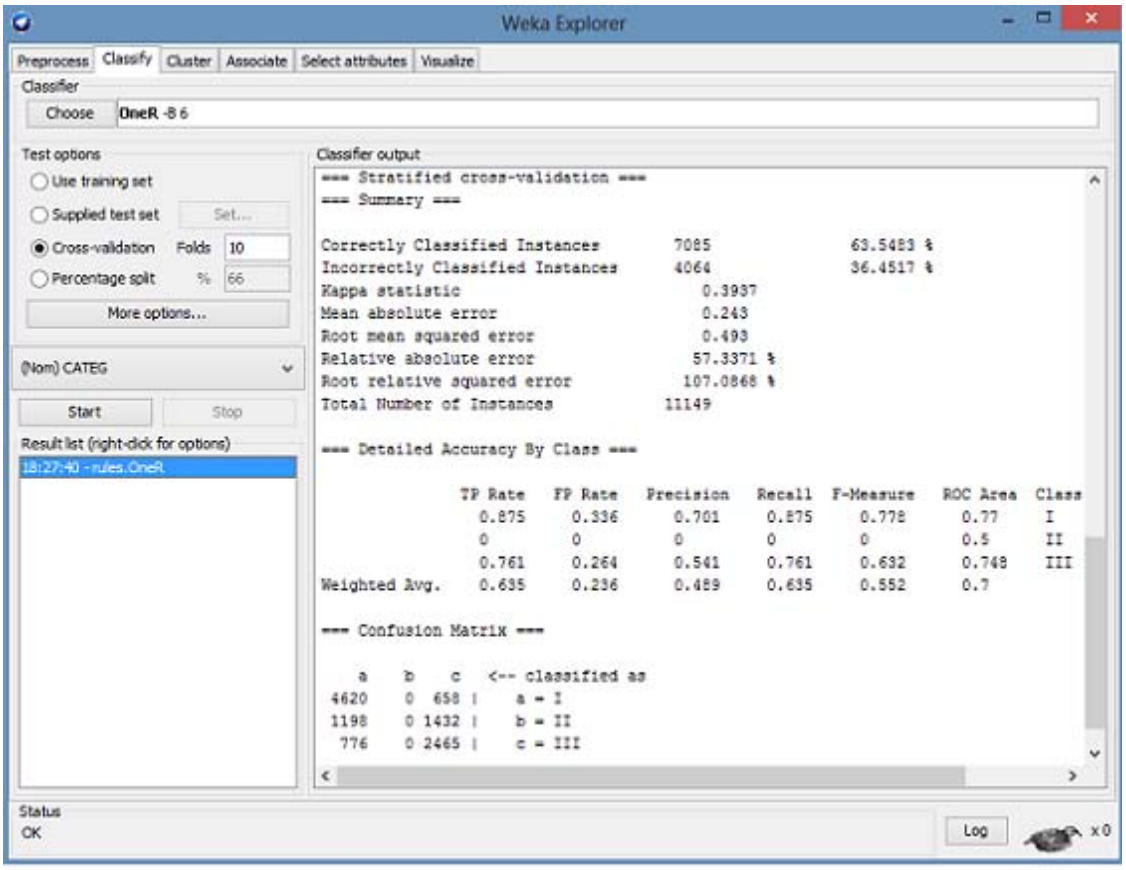


Imagen 47: Aplicación del clasificador OneR y validación cruzada.

Analizando la salida obtenemos los mismos indicadores y resultados que al usar un conjunto de entrenamiento en el modo de prueba. Es decir, la mejor predicción posible con un solo atributo es la variable POR_3 obteniendo los mismos umbrales que en el apartado anterior. Así pues por debajo del valor 9,9 las instancias se clasificarán como I por encima de ese valor como III. No encontrando ningún valor para clasificar dentro del grupo II.

La tasa de aciertos sobre el propio conjunto de entrenamiento es del 63.5483% y respecto a la matriz de confusión tenemos que de los 3.241 casos del grupo III, 2.465 se han clasificado de forma correcta y 776 se han clasificado como I y ninguno como II.

C3) Dividiendo el fichero de datos. Para realizar este apartado seleccionaremos el modo de prueba *Percentage split*. A través de este método el fichero de datos se divide en dos partes, de acuerdo al porcentaje indicado (que dejaremos en el 66%). La primera se utilizará para construir el clasificador y la segunda para evaluar su rendimiento.

Tras pulsar el botón Start aparece la información que se muestra en la Imagen 48.

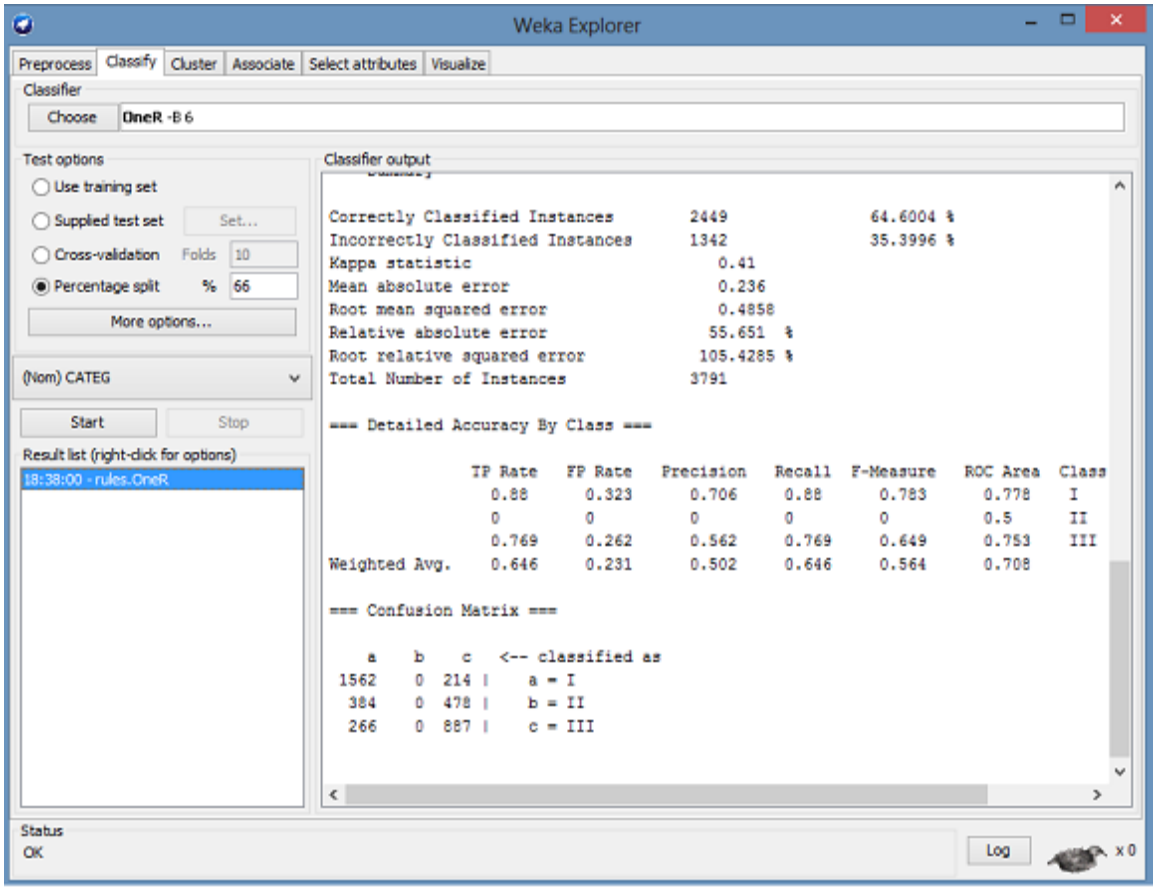


Imagen 48: Aplicación del clasificador OneR con división de ficheros

La mejor predicción posible con un solo atributo es la variable POR_3 con los siguientes umbrales:

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

POR_3:
'(-inf-9.9]' -> I
'(9.9-19.8]' -> III
'(19.8-29.7]' -> III
'(29.7-39.6]' -> III
'(39.6-49.5]' -> III
'(49.5-59.4]' -> III
'(59.4-69.3]' -> III
'(69.3-79.2]' -> III
'(79.2-89.1]' -> III
'(89.1-inf)' -> III
(7085/11149 instances correct)

Por lo tanto, por debajo del valor 9,9 los casos se clasificarán como I por encima de ese valor como III. No encontrando ningún valor para clasificar dentro del grupo II.

La tasa de aciertos sobre el propio conjunto de entrenamiento es del 64,6004% y respecto a la matriz de confusión tenemos que de los 862 casos del grupo II, 384 se han clasificado como I y 478 como III, por lo que no existe ninguno de estos casos que se haya clasificado de forma correcta.

D) Algoritmo de clasificación J48

El algoritmo J48 implementado en Weka es una versión del clásico algoritmo de árboles de decisión C4.5 propuesto por Quilan. Los árboles de decisión entran dentro de los métodos de clasificación supervisada, es decir, se tiene una variable dependiente o clase, y el objetivo del clasificador es determinar el valor de dicha clase para casos nuevos.

Para aplicar este algoritmo a nuestros datos discretizados nos situamos en la pestaña *Classify* y dentro de la sección *Classifier* pulsamos *Choose*, seguidamente seleccionamos J48 en *Trees*.

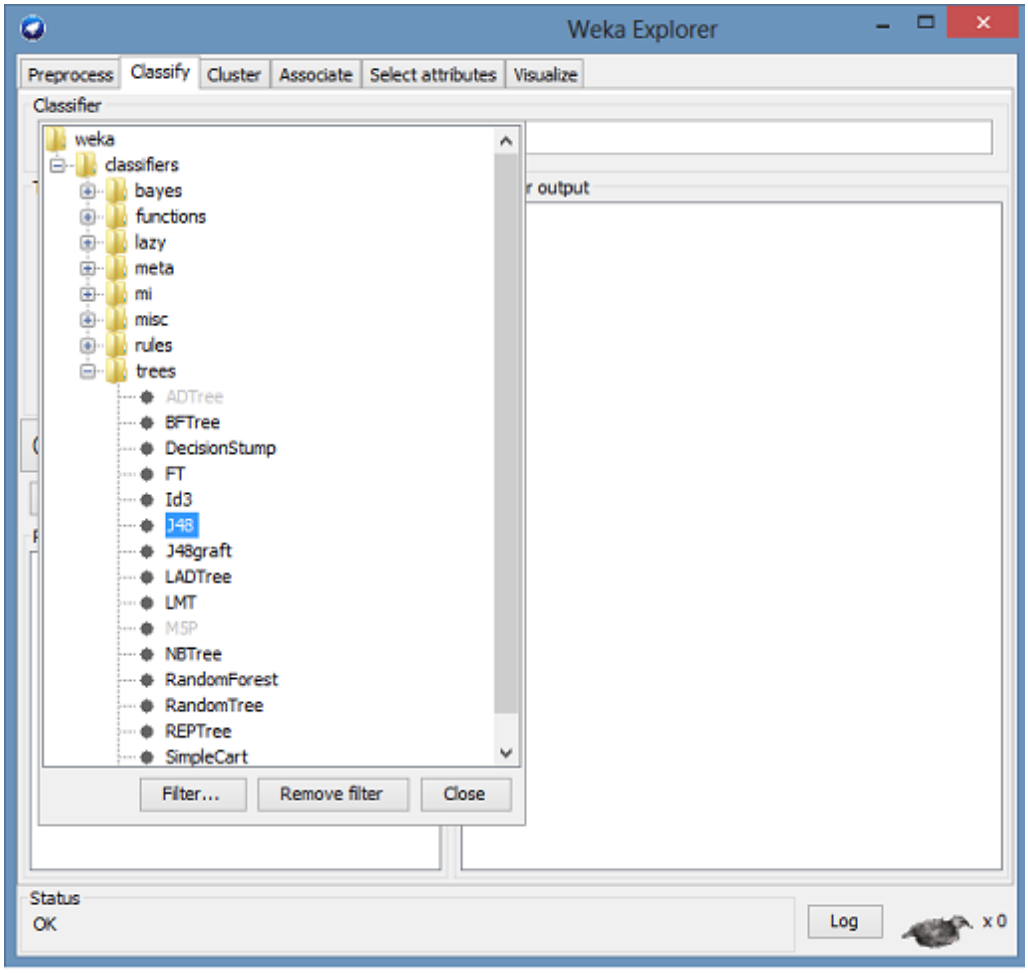


Imagen 49: Selección del algoritmo J48.

Si pulsáramos sobre la ventana que contiene el nombre del método podríamos modificar los parámetros específicos de este algoritmo. En este caso dejaremos los valores por defecto.

Una vez seleccionado este método de clasificación y para cada modo de prueba obtenemos los siguientes resultados:

D1) Usando un conjunto de entrenamiento. Seleccionamos como opción de evaluación (*Test options*) la opción *Use training set* y una vez pulsado *Start* se muestran los resultados de la Imagen 50.

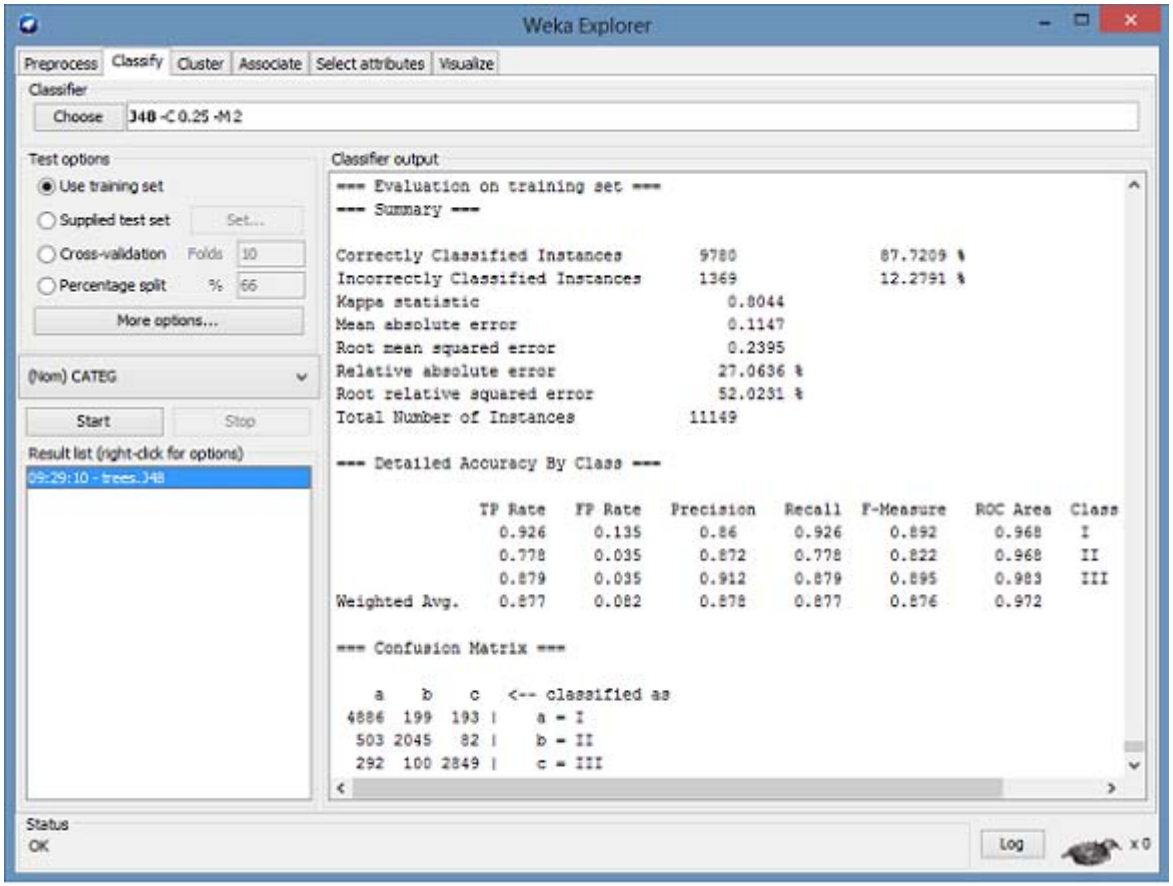


Imagen 50: Resultado de aplicar el algoritmo J48 y un conjunto de entrenamiento

Entre los resultados que se obtienen, aparte de los indicadores sobre la precisión de la clasificación, tenemos que el programa genera el árbol de decisión para nuestros datos. Parte de ese árbol se muestra a continuación.

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

J48 pruned tree
------------------

POR_2 = '(-inf-9.9]'
|   POR_3 = '(-inf-9.9]'
|   |   POR_7 = '(-inf-10]'
|   |   |   POR_4 = '(-inf-9.9]'
|   |   |   |   POR_1 = '(-inf-11.7]'
|   |   |   |   |   POR_8 = '(-inf-10]'
|   |   |   |   |   |   AZIMU = '(-inf-11.100981]': II (241.0/50.0)
|   |   |   |   |   |   AZIMU = '(11.100981-22.200872]'
|   |   |   |   |   |   |   UVER = '(-inf-23862.3]'
|   |   |   |   |   |   |   |   GLO = '(-inf-139.104]': II (0.0)
|   |   |   |   |   |   |   |   GLO = '(139.104-258.588]': II (0.0)
...
...
...
...
```

Y entre la información que se puede obtener tenemos que si POR_2 = '(-inf-9.9]', POR_3 = '(-inf-9.9]', POR_7 = '(-inf-10]', POR_4 = '(-inf-9.9]', POR_1 = '(-inf-11.7]', POR_8 = '(-inf-10]' y AZIMU = '(-inf-11.100981]' entonces las instancias se clasifican dentro del grupo II.

También podemos visualizar el árbol de forma gráfica si pulsamos el botón derecho sobre el texto t*rees.J48* de la caja *Result-list* y seleccionamos la opción *Visualize Tree.* Sin embargo y debido al gran número de ramas y hojas que se generan con nuestros datos no se visualiza de forma correcta por lo que esta forma de visualización será útil cuando el número de ramas y hojas sea bastante inferior al actual.

Respecto a los resultados de la evaluación hemos obtenido que el 87,7209% de las instancias se han clasificado correctamente mientras que el 12,2791 % lo han hecho de forma incorrecta.

El índice Kappa toma un valor de 0,8044 por lo que existe un alto grado de concordancia entre las categorías pronosticadas por el clasificador y las categorías observadas.

El error absoluto relativo es de un 27,0636% y respecto a la matriz de confusión tenemos que de los 5.278 casos del grupo I, 4.886 se han clasificado correctamente, 199 como II y 193 como III.

D2) Usando validación cruzada. Para este caso seleccionamos el método *Cross-validation* y mantenemos el número de pliegues que aparecen por defecto (10). Al pulsar el botón *Start* se genera un nuevo árbol de decisión y el resumen de los indicadores sobre la calidad de la clasificación.

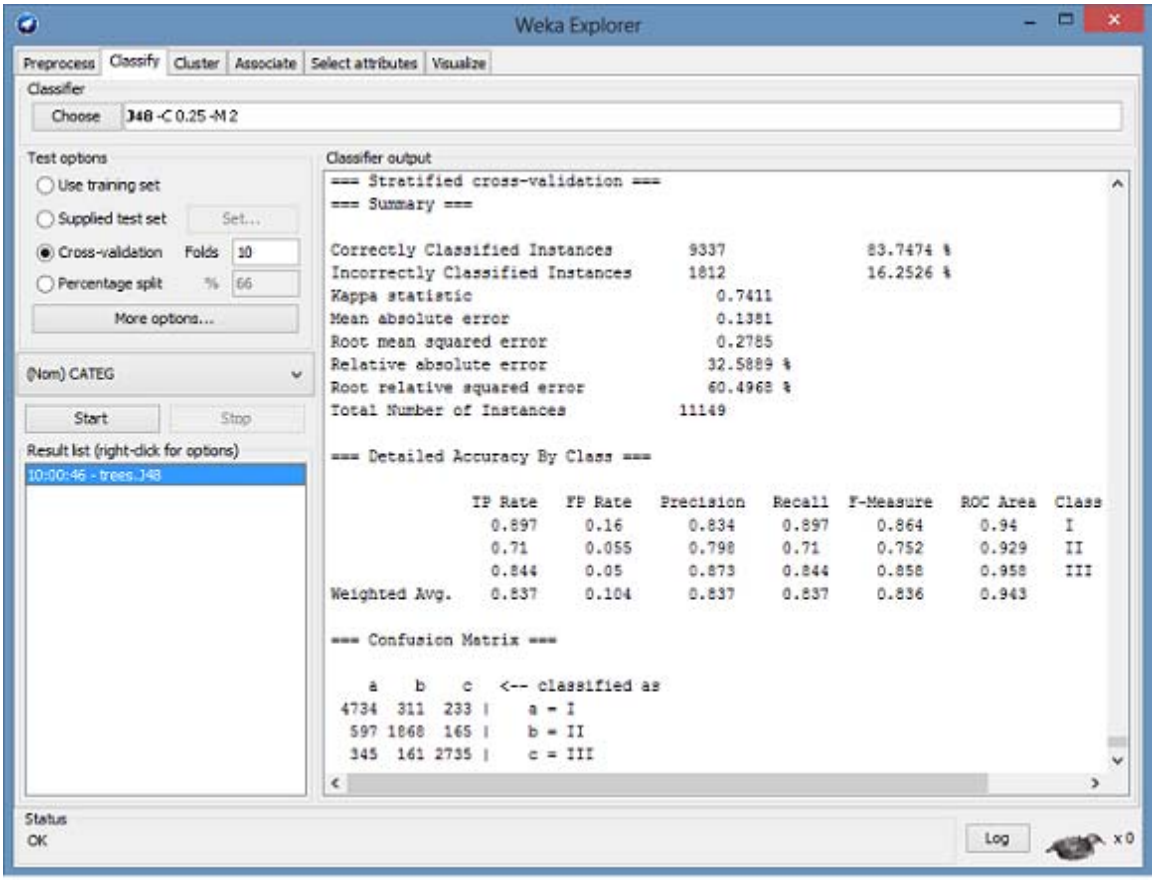


Imagen 51: Resultado de aplicar el algoritmo J48 y validación cruzada

En este caso el porcentaje de casos clasificados correctamente es del 83,7474 %, siendo el de mal clasificados el 16,2526 %. El índice Kappa toma el valor 0,7411 que es inferior al del anterior modo de prueba.

Respecto a los niveles de precisión por clase las tasas verdaderos positivos para las clases I y II son 0,897 y 0,844 respectivamente, mientras que para la clase II es de 0,71.

A través de la matriz de confusión se muestra que de 2.630 instancias del grupo II, 1.868 se han clasificado correctamente mientras que 597 se han clasificado como I y 165 lo han hecho como III.

D3) Dividiendo el fichero de datos. En este apartado seleccionaremos el modo de prueba *Percentage split*. De esta forma el fichero de datos se divide en dos partes de acuerdo al porcentaje indicado (que dejaremos en el 66%). La primera se utilizará para construir el clasificador y la segunda para evaluar su rendimiento.

Tras pulsar el botón *Start* se genera el correspondiente árbol de decisión y el conjunto de indicadores sobre la calidad de la clasificación, lo que se muestra en la Imagen 52.

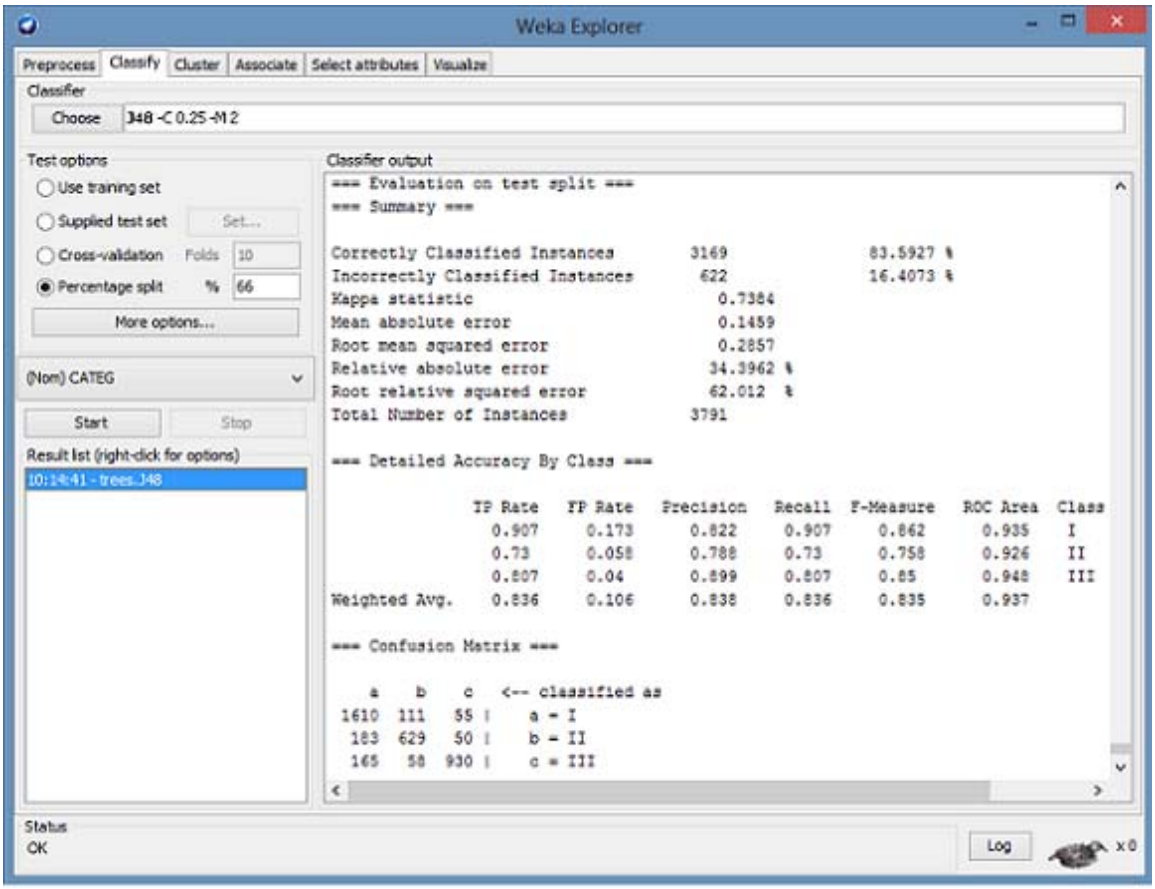


Imagen 52: Resultado de aplicar el algoritmo J48 con división del fichero de datos

En esta situación, el número de casos clasificados correctamente es del 83,5927% mientras que los clasificados incorrectamente son el 16,4073%.

El valor del índice Kappa toma un valor de 0,7384 lo que significa que existe un alto grado de concordancia entre las categorías pronosticadas por el clasificador y las categorías observadas.

Finalmente y a modo de ejemplo, la matriz de confusión muestra que de los 1.776 casos del grupo I, 1.610 se han clasificado de forma correcta, 111 lo han hecho como del grupo II y 55 como del grupo III.

4.6. Conclusiones

Los resultados obtenidos en los anteriores procesos de clasificación se resumen en la siguiente tabla:

Clasificador	Modo de prueba	Instancias bien clasificadas (%)	Instancias mal clasificadas (%)	Índice Kappa	Error absoluto
Naïve Bayes	Conjunto de entrenamiento	81,7472	18,2528	0,7137	0,1655
Naïve Bayes (mejora de precisión)	Conjunto de entrenamiento	82,6522	17,9478	0,7186	0,1646
Naïve Bayes (mejora de precisión)	Validación cruzada	81,7024	18,2976	0,7136	0,1671
Naïve Bayes (mejora de precisión)	División del fichero	81,4297	18,5703	0,7096	0,1681
Stacking	Conjunto de entrenamiento	88,0169	11,9831	0,8077	0,1244
Stacking	Validación cruzada	85,344	14,656	0,7664	0,1275
Stacking	División del fichero	85,703	14,297	0,7714	0,1295
OneR	Conjunto de entrenamiento	63,5483	36,4517	0,3937	0,243
OneR	Validación cruzada	63,5483	36,4517	0,3937	0,243
OneR	División del fichero	64,6004	35,3996	0,41	0,236
J48	Conjunto de entrenamiento	87,7209	12,2791	0,8044	0,1147
J48	Validación cruzada	83,7474	16,2526	0,7411	0,1381
J48	División del fichero	83,5927	16,4073	0,7384	0,1459

Tanto el clasificador Naïve Bayes, como el Stacking y el J48 ofrecen un porcentaje de instancias correctamente clasificadas superior al 81%, si bien, como se ha comprobado con el clasificador OneR existe un atributo (POR_3) que permite clasificar de forma correcta entorno a un 64% de las instancias.

En relación a los diferentes modos de prueba, es el basado en la muestra de entrenamiento el que ofrece mejores resultados para cada uno de los clasificadores. Esto es lógico puesto que este clasificador se evalúa en el mismo conjunto sobre el que se creó el modelo de clasificación produciendo una sobreestimación de los resultados. Sin embargo esta situación no es ratificada por el clasificador OneR donde el porcentaje de instancias bien clasificadas mediante una división del fichero (64,6004%) es superior al obtenido mediante el conjunto de entrenamiento (63,5483%).

En relación al tipo de clasificador, es el meta-clasificador Stacking el que mejores resultados ofrece, concretamente los valores de los índices Kappa para cada modo de prueba (0,8077, 0,7664 y 0,7714) son superiores a sus correspondientes modos de prueba de los otros clasificadores.

Respecto al uso de un conjunto de entrenamiento para un método Stacking hay que hacer notar que no deben usarse los mismos datos con que se entrenaron las bases porque precisamente Stacking trata de corregir sus sesgos, aprende cómo cometen errores (y pueden tener memoria del conjunto de entrenamiento). Por lo tanto es recomendable utilizar para el análisis los otros modos de prueba.

Finalmente, es el clasificador de Naïve Bayes el que peores resultados ha ofrecido, aún habiendo eliminado aquellos atributos que no son relevantes y que afectan a la calidad del método. Aún así, los resultados obtenidos por este estimador resultan de gran utilidad, ya que este clasificador ofrece una medida probabilística de la importancia de las variables que intervienen en el problema.

Bibliografía

- Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, David Scuse. WEKA Manual for versión 3.7.8. University of Waikato, 2011.
- Blanquero Bravo, Rafael. Introducción a la minería de datos. Universidad de Sevilla. Año 2010. http://rblanque.us.es
- Blanquero Bravo, Rafael. Introducción a WEKA. Universidad de Sevilla. Año 2010. http://rblanque.us.es
- García Morate, Diego. Manual de Weka. 2005.
- Ian H. Witten, Eibe Frank, and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington, MA, 3 edition, 2011.
- J.L. Cubero, F, Berzal, F. Herrera. Fundamentos de Minería de datos. Máster Oficial de la Universidad de Granada en Soft Computing y Sistemas Inteligentes. 2010.
- Hernández, J. y Ferri, C. Introducción a Weka. . Curso de Doctorado Extracción Automática de Conocimiento en Bases de Datos e Ingeniería del Software. Universitat Politécnica de Valéncia, Marzo 2006.
- Graham Williams. Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery. Springer, 2011.
- Lan Huang, David Milne, Eibe Frank, and Ian H. Witten. Learning a concept-based document similarity measure. Journal of the American Society for Information Science and Technology, 2012.
- Geoff Holmes. Developing data mining applications. In International Conference on Knowledge Discovery and Data Mining, page 225. ACM, 2012.
- J. Hernández Orallo, M.J. Ramírez Quintana, C. Ferri Ramírez. Introducción a la Minería de Datos. Pearson Prentice Hall, 2004.
- Albert Bifet, Geoff Holmes, Bernhard Pfahringer, and Eibe Frank. Fast perceptron decision tree learning from evolving data streams. In Proc 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hyderabad, India, pages 299-310. Springer, 2010.
- C. Apte. The big (data) dig, OR/MS Today, Febrero 2003. http://www.lionhrtpub.com/orms/orms-2-03/frdatamining.html
- M. Berthold, D.J. Hand. Intelligent Data Analysis: An Introduction. Springer, 1999.
- D. Hand, H. Mannila, P. Smyth. Principles of Data Mining. The MIT Press, 2001.
- T. Hastie, R. Tibshirani, J. Friedman. The elements of Statistical Learning. Springer, 2001.