



DEPARTAMENTO DE INFORMÁTICA  
UNIVERSIDAD CARLOS III DE MADRID

# Grado en Ingeniería Informática

Aprendizaje Automático  
Curso 2015-2016

## Tutorial 2: Introducción a Weka

Preprocesado y clasificación de datos con la interfaz Explorer de Weka.

9 de febrero de 2016

- El principal objetivo de estos ejercicios es familiarizarse con el entorno de Weka Explorer.
- En la página web de Weka <http://www.cs.waikato.ac.nz/ml/weka/> podéis encontrar el manual de Weka así como más documentación (tutoriales, ejemplos, etc).
- Es importante seguir el orden establecido para realizar los ejercicios correctamente.

### 1. Introducción

Weka (Waikato Environment for Knowledge Analysis) es una plataforma de software libre para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka contiene una colección de algoritmos para el análisis y el modelado predictivo.

### 2. Ejercicios

Cada ejercicio tiene una serie de pasos que han de hacerse y/o contestarse en orden.

Es necesario contestar claramente a cada una de las preguntas y subpreguntas que se formulan en estos ejercicios, indicando el ejercicio y el paso al que corresponde cada respuesta.

No debe contener capturas de pantalla de código ni capturas con resultados de texto de la interfaz de Weka. Se recomienda encarecidamente hacer tablas que sinteticen varios apartados, siempre y cuando estos apartados sean de comparación de algoritmos, comparación del mismo algoritmo con distintos ficheros o distintas evaluaciones del mismo algoritmo.

También es necesario guardar separadamente los ficheros que se modifiquen en cada ejercicio para entregarlos junto con el documento.

#### Ej. 1: Los ficheros de datos

1. Descargar el fichero de datos `badges_plain.arff` y `badges.arff`
2. Abrir los ficheros con cualquier editor y estudiar su contenido.
3. ¿Cuántos atributos de **entrada** tiene el fichero de datos? ¿De qué tipo son?
4. Podría un algoritmo de aprendizaje automático identificar esa función con los datos que hay en el fichero. ¿Por qué?

## Ej. 2: Clasificar con ZeroR

1. Ejecutar la herramienta Weka.
2. Iniciar la interfaz Explorer de la herramienta.
3. Abrir el fichero de datos `badges.arff`.
4. Seleccionar el clasificador ZeroR. En las *Test options* seleccionar *Use training set* y pulsar el botón *Start* para lanzar el clasificador. ¿Qué resultado en términos de instancias correctas ofrece este algoritmo?
5. Selecciona otro algoritmo de clasificación permitido para ese conjunto de datos. ¿Qué ocurre?
6. Repite cada uno de los pasos anteriores (1 a 5) con el fichero `badges_plain.arff`. Explica las diferencias que aparecen en cada uno de los pasos. En el punto 5 selecciona el algoritmo *trees/ID3* con el segundo fichero. ¿Qué es lo que ocurre?

## Ej. 3: Generando nuevos atributos

En Aprendizaje Automático, los datos de entrenamiento a menudo pueden mejorarse gracias a la selección o extracción de características de los datos sin procesar recopilados inicialmente. Es importante utilizar atributos adecuados para cada tarea a realizar, por lo que a veces es necesario crear características más informadas que puedan mejorar el aprendizaje.

Partiendo de lo visto en el ejercicio anterior:

1. Imagina al menos 6 atributos que te parezca que podrían ser relevantes para este problema. Estos atributos inventados se deberían poder extraer tratando adecuadamente el único atributo de entrada del ejercicio anterior. Anótalos y describe en qué consiste cada uno. ¿Por qué has elegido esos atributos?
2. Una vez inventados esos 6 atributos, abre el fichero de datos `badges1.arff` con Weka. ¿Cuántos atributos de **entrada** tiene el fichero de datos? ¿De qué tipo son?
3. ¿Qué otro tipo de información estadística se muestra sobre los atributos? Pulsa el botón “Visualize All” ¿Qué se muestra? ¿Hay algún atributo que no se visualiza? En caso de que sea así, explica por qué sucede.
4. Generar un clasificador con *ZeroR*. ¿Qué es lo que ocurre? Compara estos resultados con los obtenidos en el ejercicio anterior. ¿Tienen alguna relación? ¿Por qué?
5. Generar un clasificador con *tree/ID3*. ¿Qué es lo que ocurre? ¿Qué se podría hacer para evitar este problema con ID3?

## Ej. 4: Clasificar con ID3: resolviendo problemas

Los algoritmos de aprendizaje solo aceptan determinados tipos de atributos de entrada y de salida. Por ejemplo, el algoritmo ZeroR acepta todos los posibles valores de entrada.

1. Abre la pestaña *Classify*, selecciona el algoritmo *ZeroR*, haz click sobre el nombre y pincha en “Capabilities”, en la ventana nueva que aparece. ¿Qué información aparece en el desplegable?
2. Ahora selecciona “more”. ¿Qué información proporciona?  
Los atributos de entrada pueden modificarse a través de tareas de preprocesamiento. En los siguientes pasos vamos a modificar ciertos atributos de `badges1.arff` para que pueda clasificarse con ID3.
3. Volver a la ventana de preproceso, seleccionar el atributo *name* y borrarlo.
4. Seleccionar el filtro *Filter/unsupervised/attribute/Discretize* y aplicarlo sobre el conjunto de datos. Fijar el número de *bins* a 5. ¿Qué efecto tiene este filtro?
5. En la pestaña *Classifiers* elige ID3 y marca *Use training set* en las *Test options*. Vuelve a lanzar el clasificador.
6. ¿Cuántas instancias del conjunto de entrenamiento clasifica bien?
7. ¿Qué porcentaje de instancias clasifica bien?

8. ¿Qué crees que indica la matriz de confusión?
9. ¿Cuántas instancias de cada tipo se han clasificado mal?
10. Pulsar el botón *More Options* y seleccionar la opción *Output Predictions*. ¿Cuál es la primera instancia del conjunto de entrenamiento que se clasifica mal? ¿Por qué?
11. ¿Cómo se clasificaría la instancia “Eloisa Figueroa”? ¿Cuáles son los atributos de este nombre? ¿Qué ocurre con los valores de esta instancia si utilizas el filtro usado anteriormente?
12. Modifica el fichero original introduciendo el nombre anterior y ponle clase positiva. Recuerda, si tienes enumerados e introduces un nuevo valor tienes que especificarlo también en la definición de los valores posibles del enumerado.
13. En la pestaña *Classifiers* elige ZeroR y marca *Use training set* en las *Test options*. Vuelve a generar el clasificador.
14. ¿Cómo clasifica la instancia nueva?

### Ej. 5: Clasificar con J48 (C4.5)

1. Volver a la pestaña de preproceso y cargar en Weka el fichero de datos **badges1.arff**.
2. En la pestaña *Classifiers* elige J48 y marca *Use training set* en las *Test options*. Vuelve a generar el clasificador.
3. ¿Cuántas hojas tiene el árbol generado con J48?
4. ¿Cuántas instancias del conjunto de entrenamiento clasifica bien?
5. ¿Qué porcentaje de instancias clasifica bien?
6. ¿Cuántas instancias de cada tipo se han clasificado mal?
7. ¿Cómo se clasificaría la instancia “Eloisa Figueroa”?
8. ¿Elegirías este modelo o el generado por ID3? ¿Por qué?
9. ¿Hemos encontrado la función exacta para generar las etiquetas? ¿Por qué lo sabes?

### Ej. 6: Utilizando más atributos con J48 (C4.5)

1. Volver a la pestaña de preproceso y seleccionar el filtro *Filter/unsupervised/attribute/AddExpression* para generar un nuevo atributo que calcule el número de vocales.
2. Grabar el conjunto de datos como **badges1-2.arff**.
3. Volver a construir un clasificador con J48 con el conjunto de datos del punto anterior. En este caso tienes que seleccionar la clase “*class*” en el desplegable de la pestaña *Classify*.
4. Anota el porcentaje de instancias bien clasificadas y la matriz de confusión.
5. Haz click con el botón derecho del ratón en el modelo generado que aparece en *Result list*. Visualiza el árbol generado con *Visualize Tree*. ¿Qué indican los números que aparecen en las hojas?
6. Ir a la pestaña *Visualize*.
7. Pinchar en la gráfica que relaciona el atributo creado con la clase.
8. Aumentar el valor de *Jitter*. ¿Qué efecto tiene?
9. ¿Podrías decir cuál es el rango de vocales más común en el fichero proporcionado?
10. ¿Se te ocurre algún otro atributo relacionado que pueda aportar información?
11. Tras todos estos resultados, ¿qué características o cualidades crees que deben tener los atributos para maximizar el éxito de los algoritmos de aprendizaje automático?

## Ej. 7: Balanceado de datos, selección de características y otros filtros

1. Cargar en Weka el fichero de datos `adult-data.arff`.
2. ¿Cuántos atributos de entrada tiene este fichero? ¿Cuántas instancias de entrenamiento?
3. Ejecuta el clasificador ZeroR. Selecciona en Test Options la opción “Cross-validation” ¿Qué resultados aparecen? Explica el resultado.
4. Ahora vamos a evaluar el clasificador solamente con las instancias que figuren en el fichero `adult-test.arff`. Para ello selecciona en Test Options la opción “Supplied test set”. ¿Qué resultados aparecen? ¿Estos resultados son comparables a los anteriores? ¿Por qué?
5. Repite este procedimiento con el J48 (en lugar de ZeroR) usando cross-validation y Supplied test set. ¿Qué resultados aparecen? ¿Qué porcentaje de mejora ha obtenido respecto a los resultados del ZeroR?
6. Vuelve a la pestaña Preprocess y haz click en el atributo de salida (la clase). ¿Qué proporción de datos hay de cada clase? ¿Crees que este porcentaje es apropiado para que un algoritmo de aprendizaje automático aprenda bien?
7. Vamos a modificar las instancias de entrenamiento para que tengan un porcentaje similar entre las dos clases. Para ello selecciona **supervised/instance/Resample** cambiando el parámetro **biasToUniformClass** a 1,0 ¿Qué ocurre con el atributo de salida? ¿Ha descendido el número de ejemplos de entrenamiento?
8. Tras aplicar este filtro, evalúa de nuevo con cross-validation y supplied test set los algoritmos ZeroR y J48. ¿Qué resultados dan los algoritmos? ¿Qué resultado crees que es mejor? ¿Por qué?
9. Elimina uno o dos atributos que creas que no estén beneficiando al algoritmo de aprendizaje. ¿Cuáles has eliminado? ¿Por qué?
10. Repite la evaluación anterior ¿Qué es lo que ocurre?
11. Por último aplica el filtro de normalización **unsupervised/instance/Normalize** para los atributos numéricos. ¿Qué resultados se obtienen?
12. Después del procesamiento de datos que has realizado en este apartado, ¿crees que esto ayuda al proceso de aprendizaje? ¿Por qué?
13. ¿Cuál es el mejor resultado obtenido? Justifícalo.

## 3. Directrices para la documentación

El alumno deberá entregar una memoria en formato **PDF** que podrá tener una extensión máxima de 10 páginas, incluyendo la portada y el índice. La memoria debe contener al menos los siguientes contenidos:

- Breve descripción explicando los contenidos del documento.
- Las respuestas a cada una de las preguntas y subpreguntas que se formulan en los ejercicios del tutorial.
- No debe contener capturas de pantalla de código ni capturas con resultados de texto de la interfaz de Weka. Estos resultados se deberán mostrar adecuadamente en tablas siempre que sea posible.
- Descripción de los problemas encontrados a la hora de realizar este tutorial.
- Conclusiones y opinión personal acerca del tutorial.

## 4. Normas de entrega

El tutorial se debe realizar **obligatoriamente** en grupos de 2 personas y se entregará a través del entregador que se publicará en Aula Global **hasta las 23:55 horas del día 24 de Febrero de 2016**. El nombre del archivo comprimido debe contener los últimos 6 dígitos del NIA de los dos alumnos, ej. `tutorial2-123456-234567.zip`.

El archivo comprimido debe incluir lo siguiente:

- Una memoria en formato PDF, que deberá contener los contenidos descritos en la sección 3.
- Todos los ficheros intermedios que se han utilizado: cuando se ha creado el nuevo atributo, en el último apartado con el balanceo de datos, selección de atributos y normalización.

Se valorará la claridad de la memoria, la justificación de las respuestas a la preguntas propuestas, así como las conclusiones aportadas.

El peso de este tutorial sobre la nota final de la asignatura es de 0.4 puntos.