



ESCUELA POLITÉCNICA  
SUPERIOR DE CÓRDOBA  
Universidad de Córdoba



# UNIVERSIDAD DE CÓRDOBA ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFORMÁTICA

ESPECIALIDAD EN COMPUTACIÓN

INTRODUCCIÓN AL APRENDIZAJE AUTOMÁTICO

---

## Prácticas De IAA

---

- PRÁCTICAS -

**Autor:**

Antonio Ariza García

Enrique Galán Galán

**Profesor:**

Juan Carlos Fernández Caballero

Córdoba, 29 de mayo de 2020

Bases de datos usadas: audiology\_sof, datasetAlturaOla, wine, iris

# Índice general

<b>Índice de figuras</b>	<b>IV</b>
1. Introducción . . . . .	1
<b>1. Entregable 1</b>	<b>2</b>
1. Transformación base de datos de la UCI (Machine learning repository) a formato .arff . . . . .	2
1.1. Base de datos Dermatology . . . . .	2
2. Filtros en Weka . . . . .	8
2.1. Base de datos audiology . . . . .	8
2.2. Tres filtros no supervisados . . . . .	10
2.3. Tres filtros supervisados . . . . .	15
<b>2. Entregable 2</b>	<b>19</b>
1. Preprocesamiento conjunto de datos de altura de ola . . . . .	19
1.1. Atributos TIDE, VIS, MWD . . . . .	19
1.2. Recuperación de datos perdidos . . . . .	20
1.3. Selección de atributos . . . . .	21
1.4. Correlación . . . . .	23
2. Preprocesamiento conjunto Test . . . . .	25
2.1. Tratamiento de atributos . . . . .	25

2.2.	Normalización . . . . .	25
<b>3.</b>	<b>Entregable 3</b>	<b>27</b>
1.	Base de datos WINE . . . . .	27
1.1.	Algoritmo clasificación KNN . . . . .	28
1.2.	Algoritmo Simple Logistic . . . . .	31
<b>4.</b>	<b>Entregable 4</b>	<b>34</b>
0.1.	Ejercicio 1 . . . . .	34
0.2.	Ejercicio 2 . . . . .	38
	<b>Bibliografía</b>	<b>40</b>

# Índice de figuras

1.1. Asistente conversión Excel . . . . .	3
1.2. Abrir archivo en weka . . . . .	4
1.3. Convertidor Weka . . . . .	5
1.4. Fallo del convertidor . . . . .	6
1.5. Aspecto Fichero .arff . . . . .	7
1.6. BinaryAttributesNominal=False . . . . .	9
1.7. TransformAllValues=True . . . . .	10
1.8. Antes de aplicar filtro Normalize . . . . .	11
1.9. Después de aplicar filtro Normalize . . . . .	12
1.10. Indicaciones del filtro Remove . . . . .	13
1.11. Fichero con valores desconocidos . . . . .	14
1.12. Aplicación del filtro ReplaceMissingValues . . . . .	15
1.13. Opciones del filtro Resample . . . . .	16
1.14. Resultado de aplicar el filtro Resample . . . . .	17
2.1. DEWP Datos perdidos . . . . .	20
2.2. Atributo PRES . . . . .	21
2.3. Selecccion de atributos . . . . .	22
2.4. Logistic rendimiento . . . . .	23
2.5. Filtro Normalizacion . . . . .	26

3.1. Wine.arff antes de aplicar el filtro Normalize . . . . .	27
3.2. Wine.arff una vez normalizado . . . . .	28
3.3. Valores modificados para IBK, k=3 . . . . .	29
3.4. Estadísticas globales de nuestro clasificador . . . . .	29
3.5. Estadísticas de clasificación de cada clase . . . . .	30
3.6. Estadísticas de clasificación de cada clase . . . . .	30
3.7. Modelo de Regresión Logística obtenido al aplicar SampleLo- gistic . . . . .	31
3.8. Estadísticas de clasificación de cada clase . . . . .	33
4.1. Aplicación del 75 % de entrenamiento . . . . .	35
4.2. Árbol obtenido por el algoritmo C4.5 . . . . .	36
4.3. Reglas propias del árbol . . . . .	36
4.4. Estadísticas de clasificación de cada clase . . . . .	37
4.5. Estadísticas de clasificación de cada clase . . . . .	38
4.6. Evolución del CCR con respecto a TrainingTime . . . . .	39



## 1. Introducción

Este documento está realizado en Latex, forma parte de la asignatura Introducción al aprendizaje automático, de la escuela politécnica de Córdoba. Es una breve introducción a Weka, software utilizado para aprendizaje automático y la minería de datos, está escrito en Java.

El siguiente documento tiene como objetivo presentarse como guión y entrega de prácticas, puede utilizarse como pequeña guía a la introducción a Weka y aprendizaje automático.

# Capítulo 1

## Entregable 1

### 1. Transformación base de datos de la UCI (Machine learning repository) a formato .arff

En este apartado veremos como transformar una base da datos del repositorio UCI al formato .arff.

#### 1.1. Base de datos Dermatology

- Muchas de las bases de datos disponibles en la red se encuentran en formato .csv.
- Además de usar ficheros .arff, Weka también permite usar ficheros .csv aunque hay que hacerlo con precaución.
- Se pueden visualizar los ficheros con un editor de texto o mediante *Tools* → *ArffViewer*.

Una vez descargada la base de datos, la abrimos con excel. Entraremos en la opciones avanzadas en excel y cambiamos la configuración regional para que se utilice el “.” como separador decimal y la “,” como separador de miles. Una vez abierto el archivo, excel invoca el asistente de conversión de



## 1. TRANSFORMACIÓN BASE DE DATOS DE LA UCI (MACHINE LEARNING REPOSITORY) A FORMATO .ARFF

texto en el cual es importante resaltar la coma como separador de campos (Figura 1.1). Guardamos el fichero con formato nativo xls para tenerlos para futuras modificaciones y también lo guardamos como formato csv que si lo puede leer Weka.

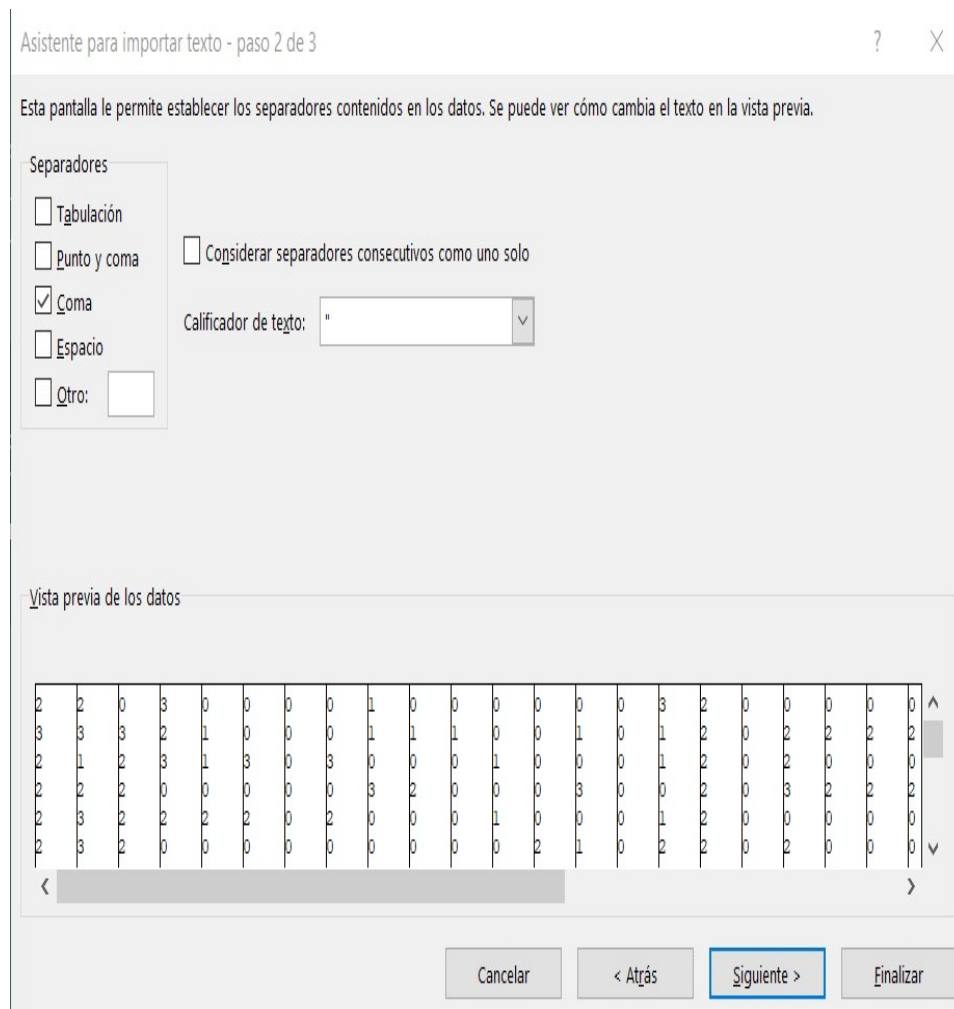


Fig. 1.1: Asistente conversión Excel

Abrimos weka, elegimos “csv” como tipo de archivo y muy importante seleccionar la opción “invoke”. (Figura 1.2). Se nos abrirá el convertidor de Weka (Figura 1.3) en donde debemos cambiar algunos parámetros.

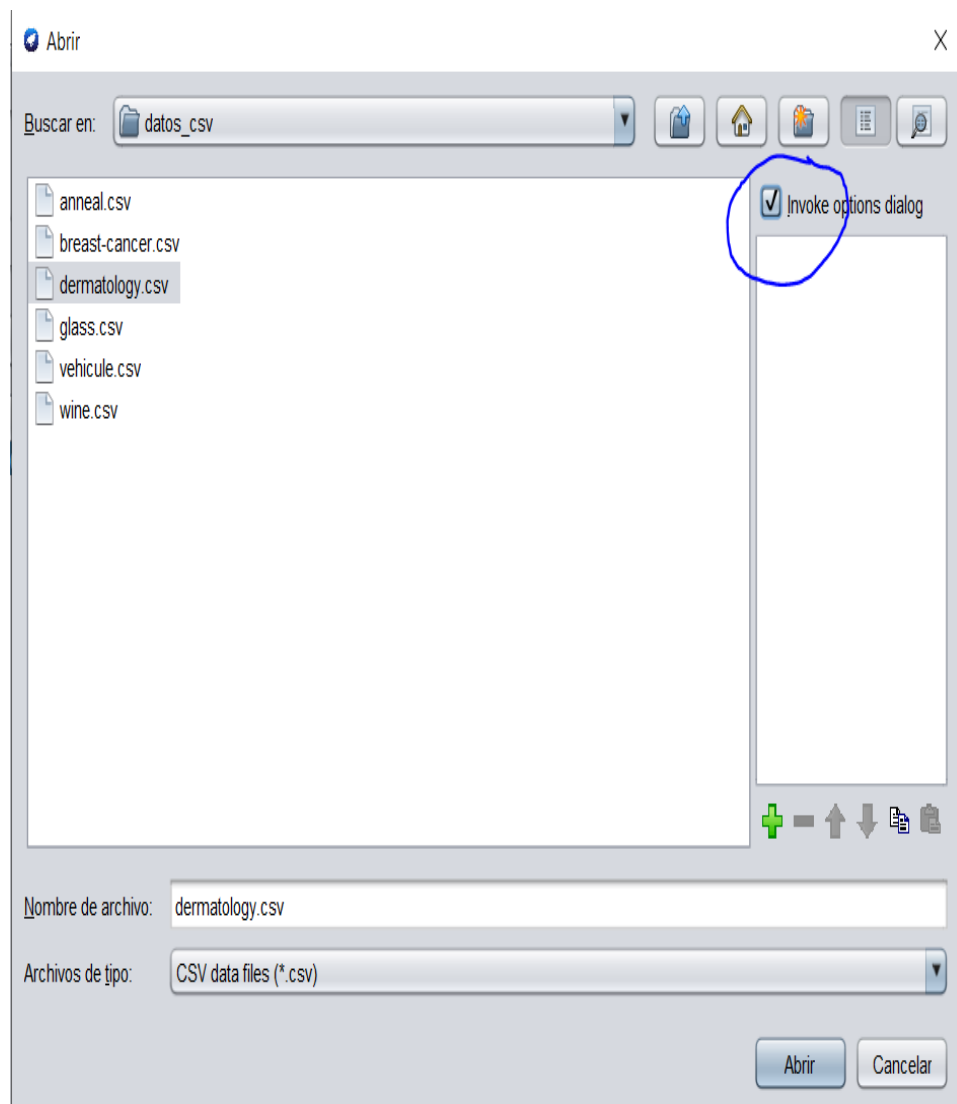


Fig. 1.2: Abrir archivo en weka

## 1. TRANSFORMACIÓN BASE DE DATOS DE LA UCI (MACHINE LEARNING REPOSITORY) A FORMATO .ARFF

---

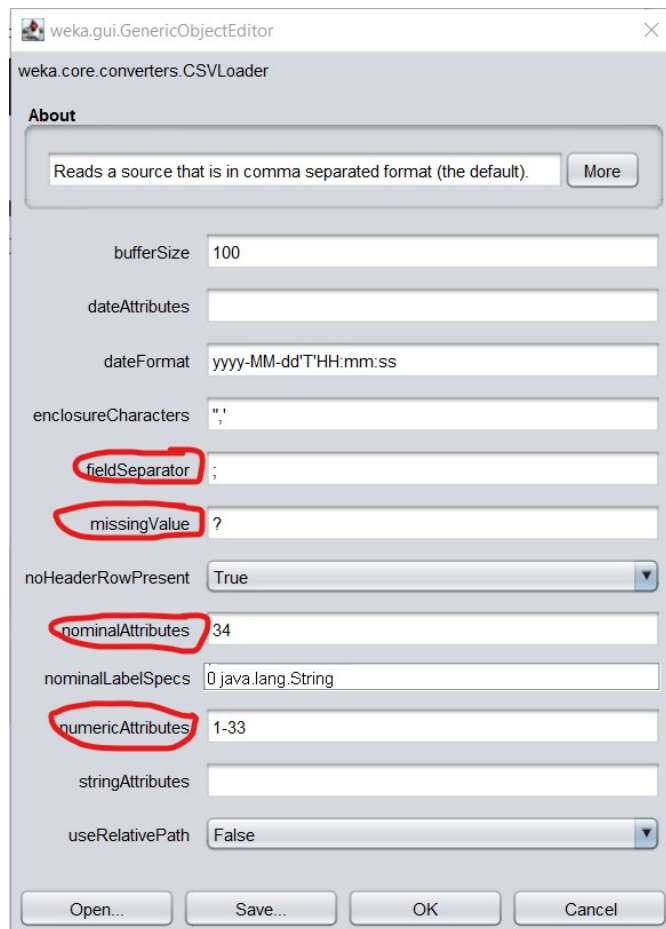


Fig. 1.3: Convertidor Weka

- FieldSeparator: Separador de campos que será “;”.
- MissingValue: Carácter que usaremos para valores perdidos.
- NoHeaderRowPresent: Decimos si el fichero no trae una fila al principio con los nombres de las variables. (En nuestro caso no la trae, decimos True).
- NominalAttributes: Lista de variables nominales.
- NumericAttributes: Lista de variables numéricas. (Se pueden poner rangos, ej:1-3,10-33).

Al aceptar comprobamos que nos dá un error (Figura 1.4), hay problemas con la versión de Weka, por lo tanto he optado a a cambiar el fichero a mano. Para cambiar los “;” es mejor hacerlo con un editor de texto como Atom o Sublime Text, dándole a la tecla Control + F se abre el buscador y nos permite reemplazar todos los “;” por “,” (con esto evitamos hacerlo uno a uno) ya que Weka utiliza la coma para la separación entre los datos. En la siguiente Figura podemos ver el aspecto de los datos del fichero .arff (Figura 1.5b), es necesario saber de que tipo son las variables (numérica o nominal) que tenemos en la base de datos y escribirlo al principio de nuestro fichero, basta con mirar en la página UCI. (Figura 1.5a). Por ejemplo elegimos la base de datos Dermatology (<https://archive.ics.uci.edu/ml/datasets/Dermatology>), pinchamos en “Data Set Description” y se nos descargará un fichero descriptivo sobre la base de datos, en el cual encontramos una descripción sobre la misma, número de atributos y tipo, número de instancias, clases.

Vemos que la última columna es la clase a predecir (variable dependiente), con 5 clases distintas (1,2,3,4,5 o U). Los valores perdidos se representan con “?” y los valores no aplicables con “-” (y deben considerarse como categoría).

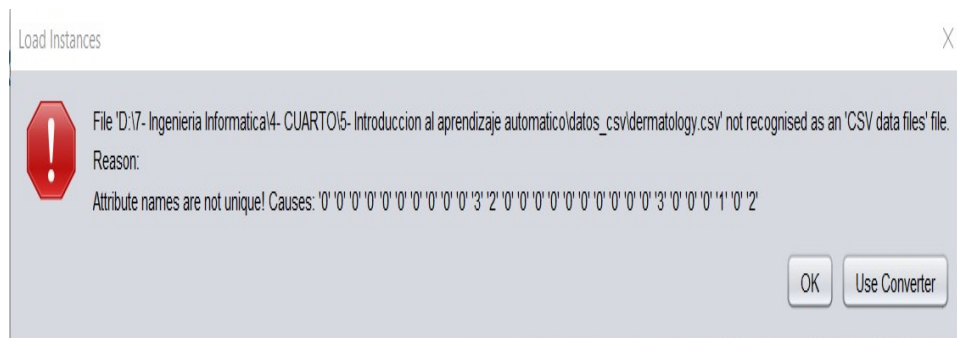


Fig. 1.4: Fallo del convertidor

## 1. TRANSFORMACIÓN BASE DE DATOS DE LA UCI (MACHINE LEARNING REPOSITORY) A FORMATO .ARFF

```
@relation breast-cancer

@attribute erythema numeric
@attribute scaling numeric
@attribute definite_borders numeric
@attribute itching numeric
@attribute koebner_phenomenon numeric
@attribute polygonal_papules numeric
@attribute follicular_papules numeric
@attribute oral_mucosal_involvement numeric
@attribute knee_and_elbow_involvement numeric
@attribute scalp_involvement numeric
@attribute family_history numeric
@attribute melanin_incontinence numeric
@attribute eosinophils_in_the_infiltrate numeric
@attribute PML_infiltrate numeric
@attribute fibrosis_of_the_papillary_dermis numeric
@attribute exocytosis numeric
@attribute acanthosis numeric
@attribute hyperkeratosis numeric
@attribute parakeratosis numeric
@attribute clubbing_of_the_rete_ridges numeric
@attribute elongation_of_the_rete_ridges numeric
@attribute thinning_of_the_suprapapillary_epidermis numeric
@attribute spongiform_pustule numeric
@attribute munro_microabscess numeric
@attribute focal_hypergranulosis numeric
@attribute disappearance_of_the_granular_layer numeric
@attribute vacuolisation_and_damage_of_basal_layer numeric
@attribute spongiosis numeric
@attribute saw_tooth_appearance_of_retes numeric
@attribute follicular_horn_plug numeric
@attribute perifollicular_parakeratosis numeric
@attribute inflammatory_mononuclear_infiltrate numeric
@attribute band_like_infiltrate numeric
@attribute Age numeric
@attribute class {1,2,3,4,5,6}

@data
2,2,0,3,0,0,0,0,1,0,0,0,0,0,3,2,0,0,0,0,0,0,0,0,3,0,0,0,1,0,55,2
3,3,3,2,1,0,0,0,1,1,1,0,0,1,0,1,2,0,2,2,2,2,1,0,0,0,0,0,1,0,8,1
2,1,2,3,1,3,0,0,3,0,0,1,0,0,0,1,2,0,2,0,0,0,2,0,2,3,2,0,0,2,3,26,3
2,3,2,2,2,2,0,2,0,0,0,1,0,0,0,1,2,0,0,0,0,0,0,2,2,3,2,3,0,0,2,3,45,3
2,3,2,0,0,0,0,0,0,0,0,0,2,1,0,2,2,0,2,0,0,0,1,0,0,0,2,0,0,0,1,0,41,2
2,1,0,2,0,0,0,0,0,0,0,0,0,3,1,3,0,0,2,0,0,0,0,0,0,0,0,2,0,0,1,0,18,5
2,2,3,3,3,3,0,2,0,0,0,2,0,0,0,2,3,0,0,0,0,0,0,0,2,2,3,2,0,0,3,3,57,3
2,2,1,0,2,0,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,0,0,0,2,0,0,2,0,0,2,0,22,4
2,2,1,0,1,0,0,0,0,0,0,0,0,0,3,2,0,2,0,0,0,0,0,0,2,0,0,2,0,0,2,0,30,4
3,3,2,1,1,0,0,0,2,2,1,0,0,0,0,0,3,2,3,2,2,2,1,1,0,0,0,0,0,1,0,20,1
2,2,0,3,0,0,0,0,0,0,0,0,0,2,0,2,2,0,0,0,0,1,0,0,0,3,0,0,0,1,0,21,2
3,3,1,2,0,0,0,0,0,1,0,0,0,2,0,3,1,0,1,0,0,0,0,0,2,0,0,0,1,0,22,2
2,3,3,0,0,0,0,0,1,1,1,0,0,1,0,0,2,1,2,1,2,3,0,2,0,0,0,0,0,2,0,10,1
2,2,3,3,0,3,0,2,0,0,2,0,0,1,1,1,0,0,0,0,2,0,3,0,3,0,0,1,3,65,3
1,1,0,1,3,0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,2,0,0,2,0,0,2,0,40,4
2,2,1,3,0,0,0,0,0,0,0,0,0,2,0,2,1,0,1,0,0,0,0,0,0,0,0,0,1,0,30,2
3,3,3,0,0,0,0,0,3,3,1,0,0,2,0,0,2,0,3,3,2,3,0,3,0,0,0,0,2,0,38,1
2,1,3,3,3,3,0,2,0,0,3,0,0,3,0,2,0,1,0,0,0,0,3,0,2,0,3,0,2,3,23,3
1,1,0,3,0,0,0,0,0,0,0,0,0,3,0,3,2,2,0,3,0,0,0,0,0,1,0,0,2,0,17,5
2,1,1,2,0,0,3,0,1,2,0,0,1,0,0,1,2,2,0,1,0,1,0,0,0,0,0,1,2,1,0,8,6
3,2,2,0,0,0,0,0,0,0,0,0,0,2,0,2,1,2,0,2,1,2,0,0,0,3,0,0,2,0,51,2
2,2,2,0,0,0,0,0,0,0,0,0,1,1,3,1,2,0,2,1,0,0,0,0,0,1,0,1,0,2,0,42,5
2,2,2,3,2,2,0,2,0,0,0,3,2,0,0,0,2,1,1,0,0,0,3,0,3,0,2,0,2,3,44,3
2,0,0,3,0,0,0,0,0,0,0,0,0,2,2,0,0,0,3,0,0,0,0,0,0,0,0,2,0,2,2,5,3
2,1,1,0,1,0,0,0,2,0,0,0,0,0,0,2,2,2,2,2,1,2,0,2,0,0,0,0,2,0,33,1
1,1,0,1,0,0,3,0,1,0,0,0,1,0,0,1,1,0,0,0,0,1,0,0,1,0,2,2,1,0,10,6
1,2,2,3,0,0,0,0,0,0,0,0,0,1,1,2,1,1,0,3,0,0,0,0,0,1,0,0,3,0,17,5
3,2,2,2,0,0,0,0,0,0,0,0,0,2,0,3,3,2,0,0,0,0,0,0,2,0,1,1,2,0,43,2
1,1,2,2,2,2,0,3,0,0,2,0,0,2,0,2,1,2,0,0,0,0,3,0,3,0,3,1,0,2,3,50,3
3,2,1,2,0,0,0,1,2,0,0,1,0,0,2,0,3,2,2,2,1,2,0,2,0,0,0,1,0,50,1
3,2,0,2,0,0,0,0,0,0,0,1,2,0,2,1,1,0,0,0,1,0,0,0,0,0,0,1,0,10,2
2,3,3,3,0,0,0,3,3,0,0,0,0,0,3,2,2,3,3,1,3,0,0,0,0,0,1,0,34,1
2,2,1,0,0,0,0,1,0,1,0,0,2,0,0,2,1,2,2,1,2,0,1,0,0,0,0,0,0,7,1
2,1,0,2,0,0,0,0,0,0,0,0,0,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,7,4
2,2,1,2,0,0,0,0,0,0,0,0,0,2,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,7,2
2,1,2,3,2,3,0,2,0,0,1,1,0,0,0,2,1,1,2,0,0,0,0,1,0,2,0,2,0,0,3,7,3
2,1,1,1,0,0,0,0,0,0,0,0,0,1,0,3,2,1,0,0,0,2,0,0,0,2,0,0,1,0,15,2
2,1,2,3,2,1,0,2,0,0,0,0,0,0,2,2,1,0,0,0,0,2,0,1,0,3,0,2,3,26,3
3,3,2,0,0,0,0,2,2,1,0,0,1,0,0,2,2,3,2,2,1,0,2,0,0,0,0,0,1,0,46,1
1,1,1,0,0,0,1,0,0,0,0,1,0,2,2,1,1,0,0,0,0,0,0,0,3,0,0,1,0,51,2
1,1,1,0,0,0,0,0,0,0,0,0,1,0,1,1,0,1,0,0,0,0,0,0,0,2,0,0,2,0,62,4
3,2,1,1,0,0,0,2,1,0,0,0,0,0,2,1,1,1,1,0,0,0,0,0,0,1,0,15,1
2,1,1,0,0,0,0,1,0,0,0,0,1,0,1,1,0,1,0,0,0,0,0,0,2,0,0,2,0,35,2
0,1,0,3,0,0,0,0,0,0,0,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,1,0,30,5
2,1,1,1,2,0,1,0,0,0,2,0,0,0,3,2,1,0,0,0,0,2,0,2,0,2,0,3,3,48,3
```

(a) Inicio fichero

(b) Final fichero

Fig. 1.5: Aspecto Fichero .arff

Para todas las bases de datos he realizado el mismo procedimiento:

- <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer>
- <https://archive.ics.uci.edu/ml/datasets/Glass+Identification>
- <https://archive.ics.uci.edu/ml/datasets/Statlog+%28Vehicle+Silhouettes%29>
- <https://archive.ics.uci.edu/ml/datasets/Wine>
- <https://archive.ics.uci.edu/ml/datasets/Zoo>

## 2. Filtros en Weka

Weka permite aplicar una gran diversidad de filtros sobre los datos, permitiendo realizar transformaciones sobre ellos de todo tipo.

- No tienen en cuenta el último atributo del dataset a la hora de hacer un tratamiento sobre los datos.
- Por defecto toman el último atributo como clase o valor numérico de salida para regresión, aplicándose el filtro a todos los patrones y atributos.

Opción Ignore Class = false

- Si queremos cargar una serie de datos a los que aplicar filtros en su totalidad, indicarlo en la opción correspondiente en el filtro.

Opción Ignore Class = true

### 2.1. Base de datos audiology

Cargue la base de datos audiology. Pruebe todas las combinaciones posibles para pasar los atributos nominales a binarios, según los detalles proporcionados en la transparencia numero 8, mostrando en cada caso un ejemplo de cómo quedarían los nominales con 2 valores y con 3 o más valores usando esas combinaciones.

- BinaryAttributesNominal=False

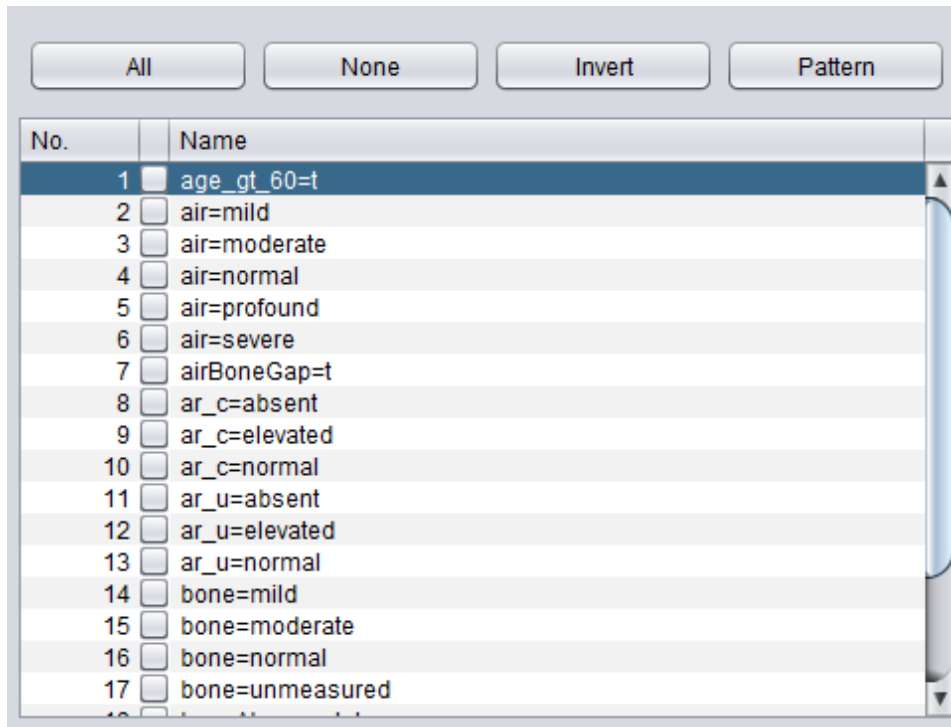


Fig. 1.6: BinaryAttributesNominal=False

Como vemos en la figura superior, los atributos nominales con dos etiquetas como es el caso de `agegt60` se ha transformado en un solo atributo binario cuyo nombre coincide con el de una de sus etiquetas en este caso `t(true)` quedando con el nombre de `agegt60=t`, aquellos que en este atributo posean un valor de 1 significara que tienen un valor true, para el atributo anterior, sin embargo aquellos que tengan un valor de 0 tendrían un valor de false. Muy distinto es el caso de los atributos nominales que poseen tres o más etiquetas como es el caso del atributo `air`, que posee hasta cinco etiquetas, en este caso se crea un atributo para cada etiqueta indicando en cada caso con uno o cero la presencia o no de este atributo.

- TransformAllValues=True

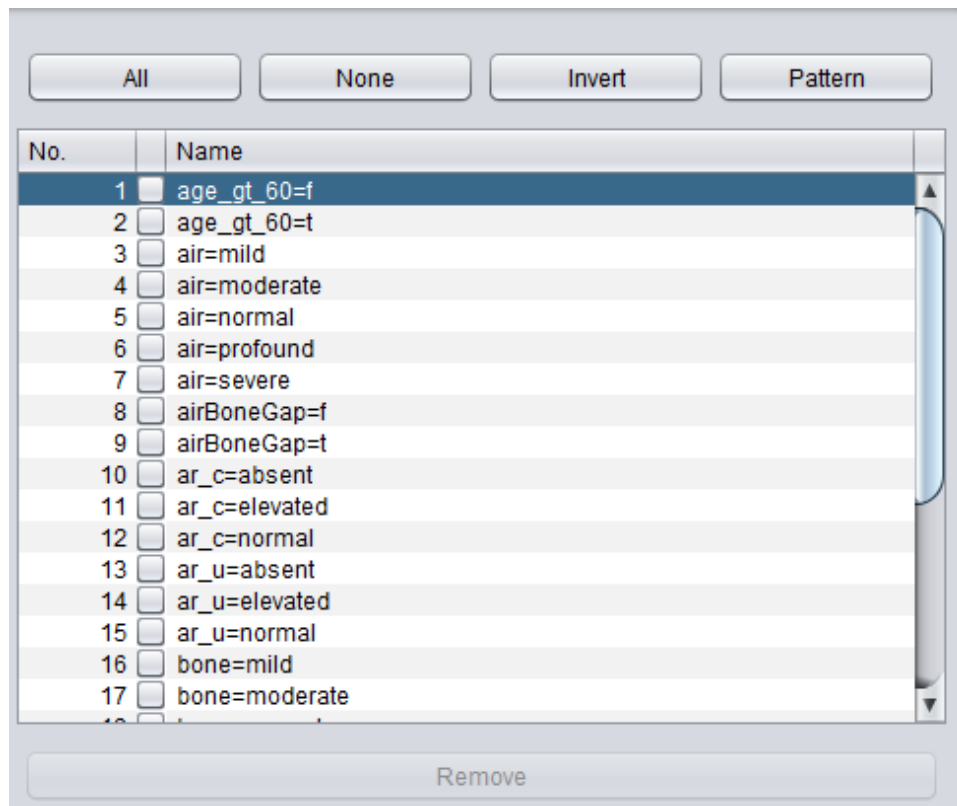


Fig. 1.7: TransformAllValues=True

En este último caso vemos que tanto aquellos atributos con dos etiquetas(agegt60) que poseían las etiquetas t(true) y f(false) han sido transformados a dos atributos binarios, uno con valor de true y otro con valor de false indicando con uno o cero la presencia o no del mismo, de igual modo aquellos con tres o mas etiquetas quedan como lo hemos visto antes transformándose en tantos atributos binarios como etiquetas tuviera el atributo nominal.

## 2.2. Tres filtros no supervisados

Elija tres filtros no supervisados de los que aparecen listados, explíquelos y describa como quedan los datos antes y después al aplicarlos sobre una o varias bases de daros de las indicadas en moodle ya en formato .arff



- Normalize: Normaliza todos los datos de manera que el rango de los datos pase a ser [0,1]. Para normalizar un vector se utiliza la fórmula:

$$X(i) = \frac{x(i)}{\sqrt{\sum_{i=1}^n x(i)^2}}$$

En este caso lo vamos a aplicar a nuestra base de datos iris.arff que posee cuatro atributos numéricos y un atributo clasificador.

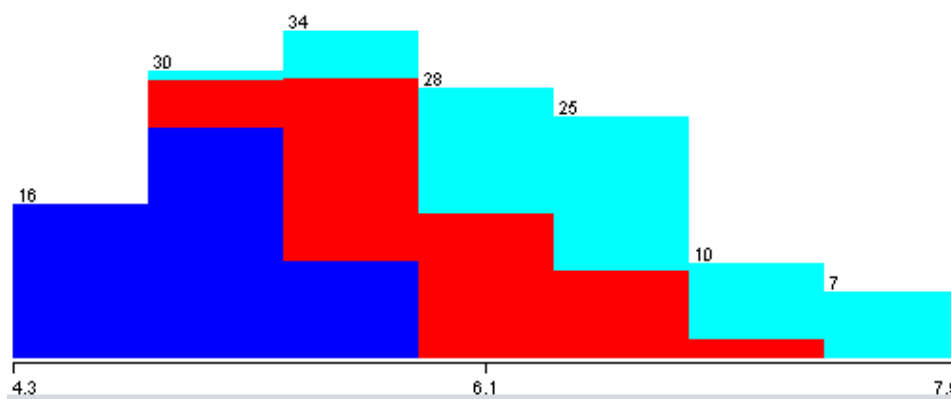
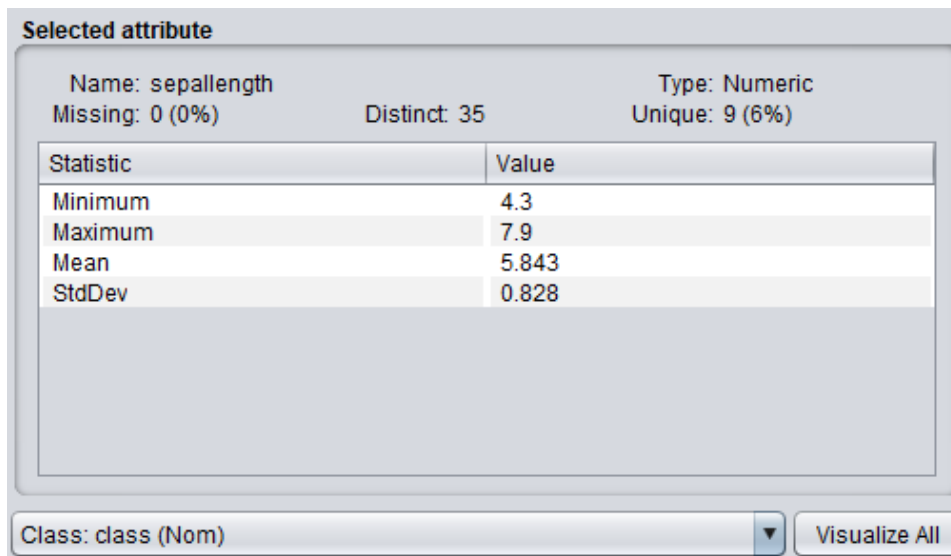


Fig. 1.8: Antes de aplicar filtro Normalize

Como vemos en la figura superior, en la cual aún no hemos aplicado el

## CAPÍTULO 1. ENTREGABLE 1

---

filtro Normalize, el atributo sepallenght de tipo numérico nos muestra como se encuentra repartidos los valores en función de su clase, pero siempre en un rango de entre 4,3 y 7,9. Sin embargo una vez aplicado el filtro Normalize este rango quedará normalizado entre los valores 0 y 1.

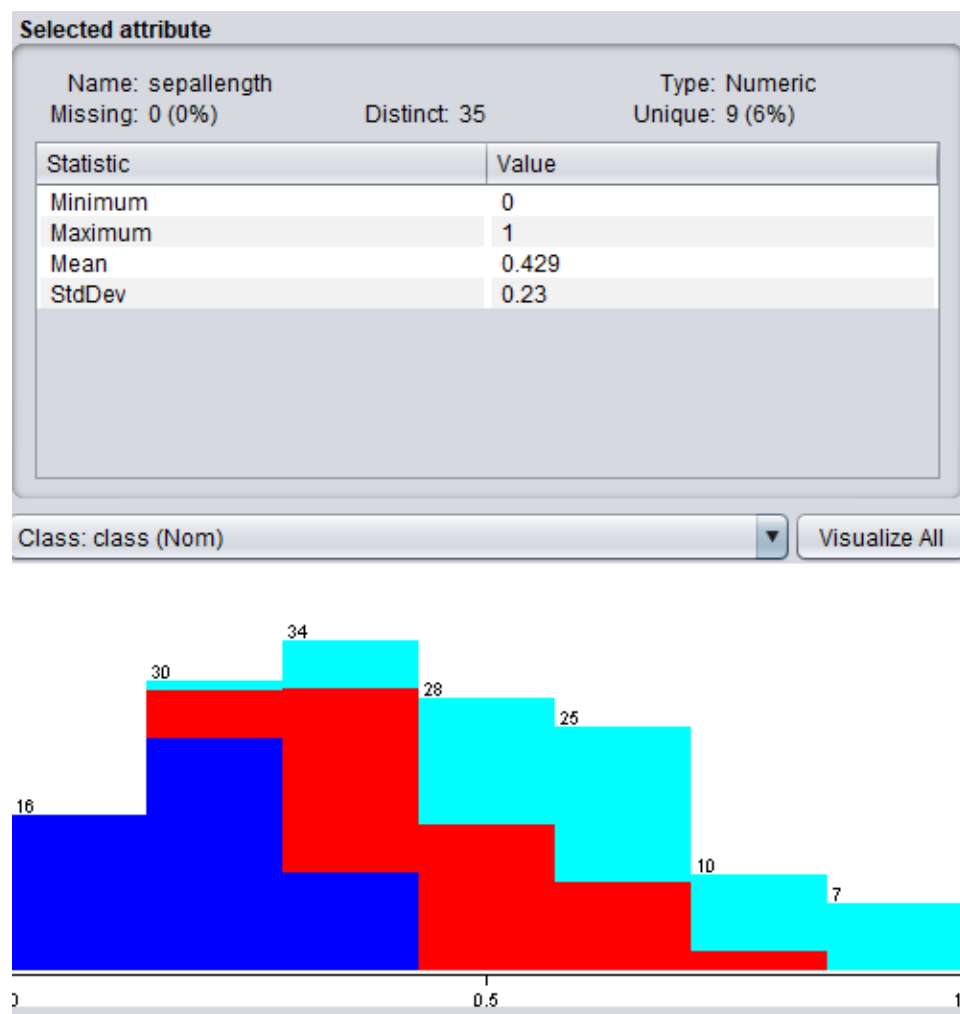


Fig. 1.9: Después de aplicar filtro Normalize

- Remove: Borra un conjunto de atributos del fichero de datos, para ello debemos de indicarle el índice de los atributos que queremos borrar de nuestro fichero de datos. A continuación se indica como hacerlo:

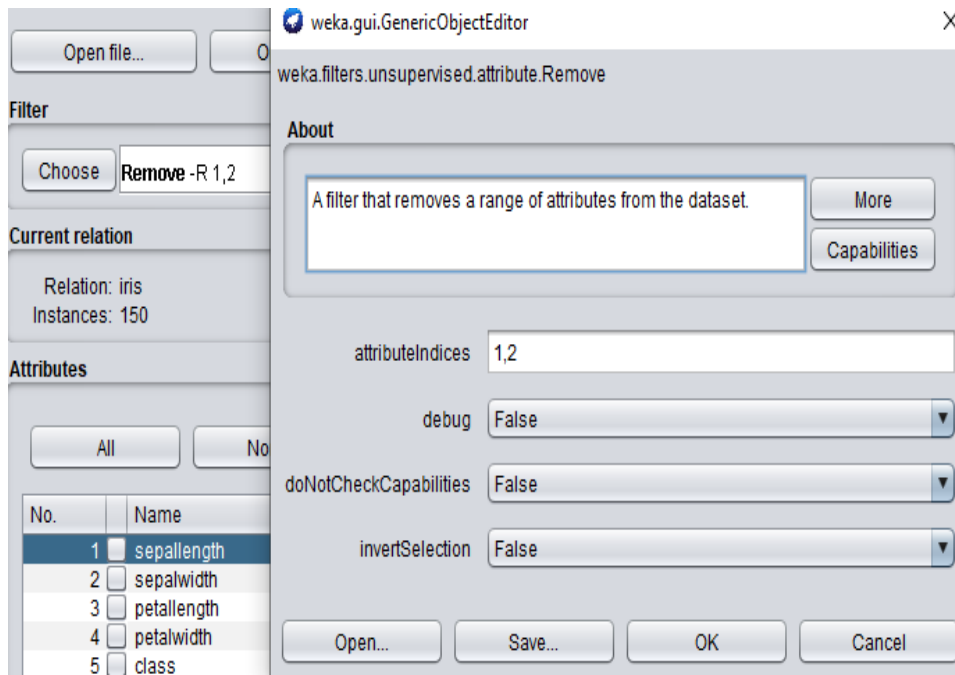


Fig. 1.10: Indicaciones del filtro Remove

Es en el apartado “attributeIndices” donde indicamos los índices de los atributos que queremos borrar, podemos indicar los índices que queremos borrar o bien separados por comas o bien separados por guión lo que indicara que queremos borrar un rango de atributos.

- **ReplaceMissingValues:** Reemplaza todos los valores indefinidos por la moda en el caso de que sea un atributo nominal o la media aritmética si es un atributo numérico. Para este caso hemos cogido el fichero de datos iris.arff y lo hemos modificado eliminando tres de sus valores como aparece a continuación.




Relation: iris

No.	1: sepallength	2: sepalwidth	3: petallength	4: petalwidth	5: class
	Numeric	Numeric	Numeric	Numeric	Nominal
1	5.1	3.5	1.4	0.2	Iris-s...
2	4.9	3.0	1.4	0.2	Iris-s...
3	4.7	3.2	1.3	0.2	Iris-s...
4	4.6	3.1	1.5	0.2	Iris-s...
5	5.0	3.6	1.4	0.2	Iris-s...
6	5.4	3.9	1.7	0.4	Iris-s...
7	4.6	3.4	1.4	0.3	Iris-s...
8	5.0	3.4		0.2	Iris-s...
9	4.4	2.9	1.4	0.2	Iris-s...
10	4.9	3.1	1.5		Iris-s...
11	5.4	3.7	1.5	0.2	Iris-s...
12	4.8	3.4	1.6	0.2	Iris-s...
13	4.8	3.0	1.4	0.1	Iris-s...
14	4.3	3.0		0.1	Iris-s...
15	5.8	4.0	1.2	0.2	Iris-s...

Fig. 1.11: Fichero con valores desconocidos

En la siguiente figura se muestra como al aplicar el filtro ReplaceMissingValues los valores se han sustituido por la media de cada atributo ya que ambos son numéricos.

 Viewer

Relation: iris-weka.filters.unsupervised.attribute.ReplaceMissingValues

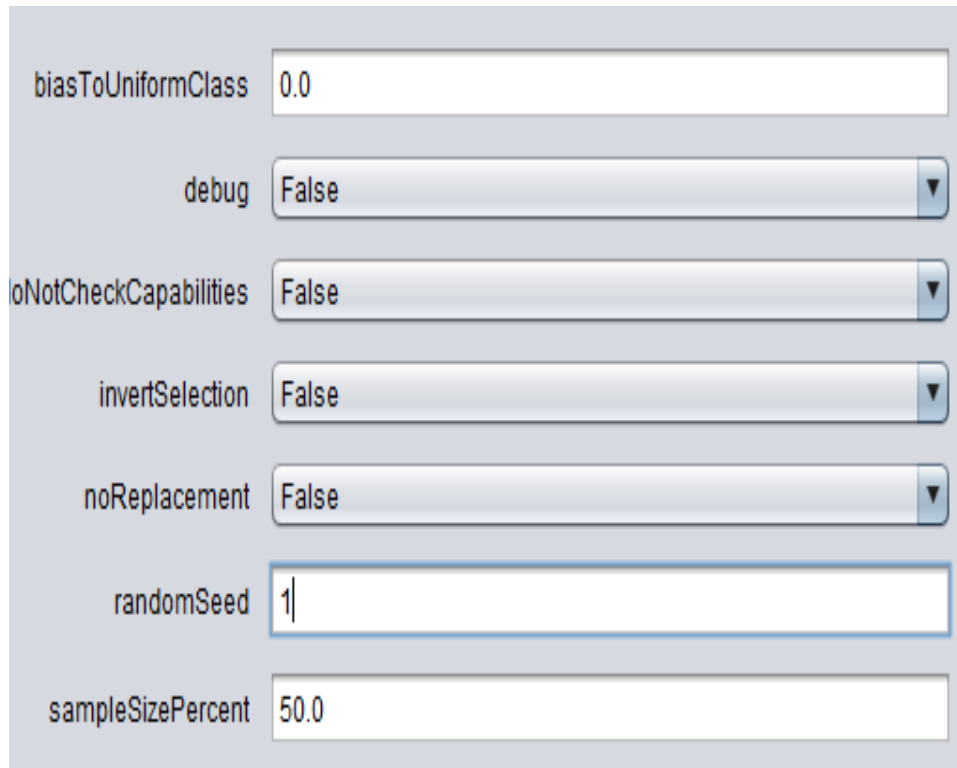
No.	1: sepallength	2: sepalwidth	3: petallength	4: petalwidth	5: class
	Numeric	Numeric	Numeric	Numeric	Nominal
1	5.1	3.5	1.4	0.2	Iris-s...
2	4.9	3.0	1.4	0.2	Iris-s...
3	4.7	3.2	1.3	0.2	Iris-s...
4	4.6	3.1	1.5	0.2	Iris-s...
5	5.0	3.6	1.4	0.2	Iris-s...
6	5.4	3.9	1.7	0.4	Iris-s...
7	4.6	3.4	1.4	0.3	Iris-s...
8	5.0	3.4	3.7918918...	0.2	Iris-s...
9	4.4	2.9	1.4	0.2	Iris-s...
10	4.9	3.1	1.5	1.206040...	Iris-s...
11	5.4	3.7	1.5	0.2	Iris-s...
12	4.8	3.4	1.6	0.2	Iris-s...
13	4.8	3.0	1.4	0.1	Iris-s...
14	4.3	3.0	3.7918918...	0.1	Iris-s...

Fig. 1.12: Aplicación del filtro ReplaceMissingValues

### 2.3. Tres filtros supervisados

Elija tres filtros no supervisados de los que aparecen listados, explíquelos y describa como quedan los datos antes y después al aplicarlos sobre una o varias bases de datos de las indicadas en moodle ya en formato .arff

- Resample: Produce una submuestra aleatoria de un conjunto de datos utilizando muestreo con reemplazo o sin reemplazo. El conjunto de datos original debe caber completamente en la memoria. Se puede especificar el número de instancias en el conjunto de datos generado. El conjunto de datos debe tener un atributo de clase nominal. De lo contrario, use la versión sin supervisión. El filtro se puede hacer para mantener la distribución de la clase en la submuestra o para sesgar la distribución de la clase hacia una distribución uniforme.



The image shows a user interface for configuring the 'Resample' filter. It consists of several rows, each with a label on the left and a control on the right. The controls include text input fields and dropdown menus. The labels and their corresponding values are: 'biasToUniformClass' with '0.0', 'debug' with 'False', 'doNotCheckCapabilities' with 'False', 'invertSelection' with 'False', 'noReplacement' with 'False', 'randomSeed' with '1', and 'sampleSizePercent' with '50.0'. The 'randomSeed' field is currently selected with a blue border.

Option	Value
biasToUniformClass	0.0
debug	False
doNotCheckCapabilities	False
invertSelection	False
noReplacement	False
randomSeed	1
sampleSizePercent	50.0

Fig. 1.13: Opciones del filtro Resample

Como vemos en la figura de arriba estas son las distintas opciones que nos plantea el filtro Resample, en este caso en particular hemos decidido el generar una nueva muestra aleatoria que siguiera la misma distribución que los datos de entrada y además tal como vemos en la ultima opción `sampleSizePercent` en la cual indicamos un cincuenta, esto quiere decir que el tamaño de nuestra muestra generada será del cincuenta por ciento de nuestra muestra inicial. En la siguiente figura mostramos los resultados de su aplicación.

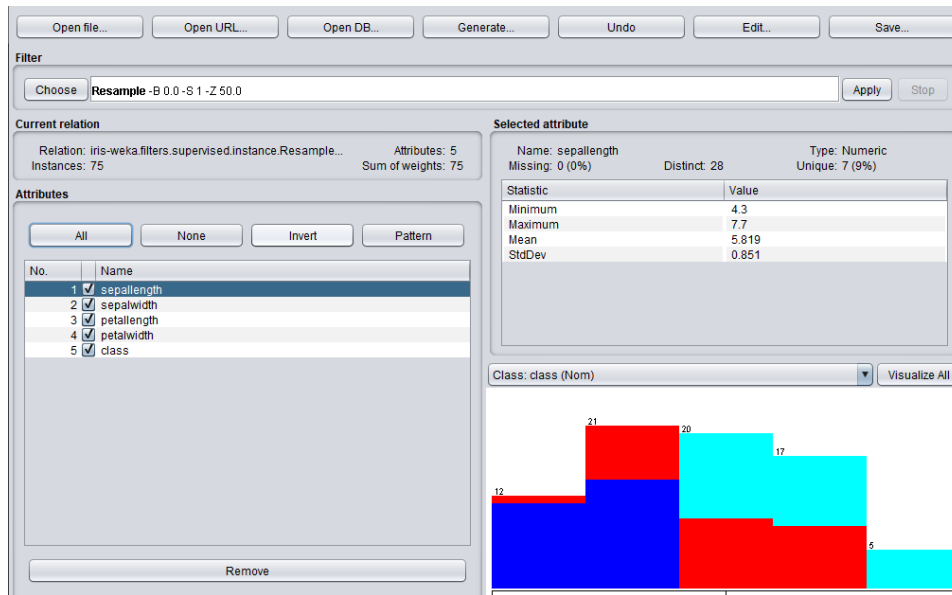


Fig. 1.14: Resultado de aplicar el filtro Resample

- SpreadSubsample: Produce una submuestra aleatoria de un conjunto de datos. El conjunto de datos original debe caber completamente en la memoria. Este filtro le permite especificar la "dispersión" máxima entre la clase más rara y la más común. Por ejemplo, puede especificar que haya como máximo una diferencia de 2: 1 en las frecuencias de clase. Cuando se usa en modo por lotes, los lotes posteriores NO se vuelven a muestrear.
- Discretize: Discretiza un conjunto de valores numéricos en rangos de datos. Como parámetros toma los índices de los atributos discretizar (attribute indices) y el número de particiones en que queremos que divida los datos (bins). Si queremos que las particiones las realice por la frecuencia de los datos y no por el tamaño de estas tenemos la opción useEqualFrequency. Si tenemos activa esta última opción podemos variar el peso de las instancias para la definición de los intervalos con la opción DesiredWeightOfInstancesPerInterval. Si, al contrario tenemos en cuenta el número de instancias para la creación de intervalos po-

demostramos usar `findNumBins` que optimiza el procedimiento de confección de los mismos. Otras opciones son `makeBinary` que convierte los atributos en binario e `invertSelection` que invierte el rango de los atributos elegidos.



## Capítulo 2

# Entregable 2

- El preprocesamiento es uno de los procesos más importantes en el flujo de acciones sobre un conjunto de datos. Es determinante para la obtención de modelos con buen rendimiento.
- Cada conjunto de datos necesita un preprocesamiento concreto diferente del realizado a otros.

### 1. Preprocesamiento conjunto de datos de altura de ola

Describa las operaciones de procesamiento que ha realizado sobre la base de datos proporcionada y como queda la base de datos al final ya procesada.

#### 1.1. Atributos TIDE, VIS, MWD

Eliminaremos aquellos atributos que tengan más del 40 % de datos perdidos. Como podemos comprobar los atributos TIDE, VIS, MWD no aportan nada a nuestro modelo, ya que todos sus valores son nulos. Eliminaremos estos tres atributos de la base de datos, quedándonos con los 15 restantes. Podemos utilizar el filtro Remove explicado en el capítulo anterior. (1.10)

El atributo DEWP tiene 33 % de los valores perdidos (2.1), por lo tanto también hemos decidido eliminarlo, pues tendríamos que reemplazar todos esos valores por la media e insertaríamos muchos datos iguales en la base de datos.

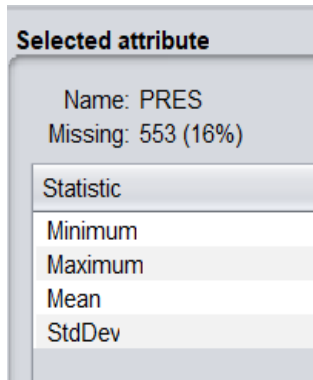
Selected attribute	
Name: DEWP Missing: 1189 (33%)	Distinct: 203 Type: Numeric Unique: 22 (1%)
Statistic	Value
Minimum	-9.6
Maximum	14.6
Mean	6.221
StdDev	4.267

Fig. 2.1: DEWP Datos perdidos

### 1.2. Recuperación de datos perdidos

Existen muchas técnicas para la recuperación de datos perdidos.

- Reemplazar por la media del conjunto de datos. También se puede reemplazar por la mediana o la moda dependiendo del tipo de atributo. Es un poco más justa cuando se emplean patrones de la misma clase.
- Regresión entre atributos (sin datos perdidos).
- Mediante técnicas de Machine Learning.



Selected attribute	
Name:	PRES
Missing:	553 (16%)
Statistic	
Minimum	
Maximum	
Mean	
StdDev	

Fig. 2.2: Atributo PRES

El atributo PRES tiene el 16 % de datos perdidos (Fig2.2). En nuestro caso hemos utilizado el reemplazo por la media, para hacer esto utilizamos el filtro “ReplaceMissingValues” explicado en el capítulo anterior, (Fig1.11) este filtro no calcula la media dependiendo del atributo clase si no que le asigna la misma media a todos. Al utilizar el filtro veremos como el porcentaje “Missing” se pone a 0 %.

### 1.3. Selección de atributos

Consiste en obtener una representación reducida del conjunto de datos que preserve información importante de la base de datos. Objetivos:

- Reducir la complejidad del problema eliminando atributos irrelevantes o redundantes.
- Aumentar el rendimiento de los modelos.
- Acelerar el proceso de aprendizaje.
- Reducción del sobreajuste.

Aunque en Weka hay muchas formas de seleccionar atributos (como análisis de correlaciones), nos centraremos en el método mediante búsqueda + evaluación. Seleccionamos como: “*Attributeevaluator*” → *CfsSubsetEval* y como “*SearchMethod*” → *BestFirst* En la (Figura2.3) comprobamos como nos selecciona los atributos 2,4,5,7,8,9,10,11,12.

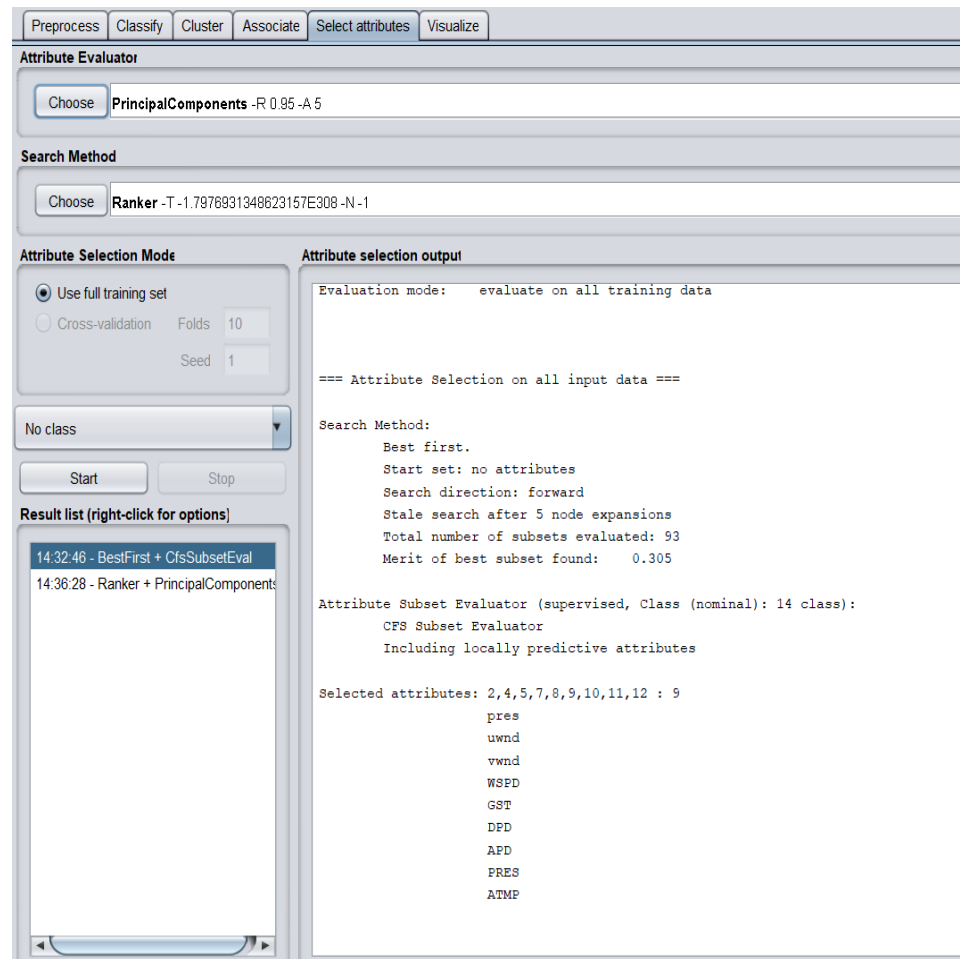


Fig. 2.3: Selección de atributos

Volviendo a nuestro conjunto de entrenamiento procedemos a utilizar el filtro “Remove” para dejar solo los atributos indicados, para ellos copiamos y pegamos los atributos e invertimos la selección para que nos elimine los no seleccionados (en este caso eliminaremos los atributos AIR, RHUM, WDIR, WTMP). Una vez tenemos los atributos procedemos a realizar un experimento con un clasificador “Logistic” y la configuración por defecto para comprobar si hemos mejorado el rendimiento y los atributos eliminados no aportaban mucha información. (2.4)

## 1. PREPROCESAMIENTO CONJUNTO DE DATOS DE ALTURA DE OLA

---

=== Summary ===

Correctly Classified Instances	2069	71.6909 %
Incorrectly Classified Instances	817	28.3091 %
Kappa statistic	0.6098	
Mean absolute error	0.1895	
Root mean squared error	0.3138	
Relative absolute error	51.7921 %	
Root relative squared error	73.2455 %	
Total Number of Instances	2886	

(a) Antes

=== Summary ===

Correctly Classified Instances	2091	72.4532 %
Incorrectly Classified Instances	795	27.5468 %
Kappa statistic	0.6202	
Mean absolute error	0.1898	
Root mean squared error	0.3132	
Relative absolute error	51.8854 %	
Root relative squared error	73.1028 %	
Total Number of Instances	2886	

(b) Despues

Fig. 2.4: Logistic rendimiento

### 1.4. Correlación

El rendimiento de algunos algoritmos puede deteriorarse si dos o más variables están estrechamente relacionadas. Eliminar algunas de las variables pueden mejorar el rendimiento del modelo. Para obtener la matriz de correlaciones nos situamos en la pestaña de selección de atributos y seleccionamos como: “*Attributeevaluator*” → *PrincipalComponents* y como “*SearchMethod*” → *Ranked*

## CAPÍTULO 2. ENTREGABLE 2

---

	pres	uwnd	vwnd	WSPD	GST	DPD	APD	PRES	ATMP
pres	1	0.14	-0.2	-0.34	-0.37	-0.26	-0.43	0.93	0.26
uwnd	0.14	1	-0.17	-0.07	-0.17	-0	-0.03	0.05	0.17
vwnd	-0.2	-0.17	1	0.31	0.31	0.07	0.13	-0.18	-0.01
WSPD	-0.34	-0.07	0.31	1	0.99	0.01	0.1	-0.31	-0.17
GST	-0.37	-0.07	0.31	0.99	1	0.04	0.14	-0.34	-0.2
DPD	-0.26	-0	0.07	0.01	0.04	1	0.71	-0.27	-0.33
APD	-0.43	-0.03	0.13	0.1	0.14	0.71	1	-0.44	-0.44
PRES	0.93	0.05	-0.18	-0.31	-0.34	-0.27	-0.44	1	0.28
ATMP	0.26	0.17	-0.01	-0.17	-0.2	-0.33	-0.44	0.28	1

Tabla 2.1: Matriz de correlación

Comprobando la matriz de correlación (Tabla 2.1) nos damos cuenta como existe una relación entre el atributo GST y WSPD. Mirando el resto de los atributos hemos decidido eliminar el atributo WSPD, debemos tener cuidado porque este análisis puede llevar a ilusiones o relaciones falsas. Eliminaremos el atributo y haremos algunas pruebas con el conjunto de test en el cual también debemos eliminar los mismos atributos que en el conjunto de entrenamiento (Mirar el siguiente punto relacionado con el conjunto de test). Al realizar las pruebas obtenemos un 72,35 % de acierto en clasificación, anteriormente obtuvimos un 72,45 %, aunque hemos perdido 0,1 es señal de que las variables están relacionadas y no aportan mucho al conjunto de datos. Continuando con este procedimiento y obteniendo otra vez la matriz de correlaciones podemos continuar eliminando algunos atributos más. En nuestro caso hemos eliminado:

- WSPD
- PRES
- uwnd

## 2. Preprocesamiento conjunto Test

El proceso de procesamiento del conjunto test es prácticamente similar al del conjunto de entrenamiento, con algunas pequeñas diferencias, variando únicamente el proceso de normalización de atributos y el reemplazamiento de los atributos perdidos.

### 2.1. Tratamiento de atributos

Todos los atributos eliminados en el conjunto de datos de entrenamiento deben ser eliminados también en el conjunto de test, tanto los eliminados al principio como los eliminados en el proceso de selección de atributos. También hay que reemplazar los datos perdidos como hicimos con el atributo PRES por ejemplo, este reemplazamiento debe ser por los valores usados en train (podemos hacer uso del filtro “ReplaceMissingWithUserConstant”).

### 2.2. Normalización

Para normalizar los datos del conjunto de test debemos obtener todos los mínimos y máximos de cada atributo que podemos ver pinchando encima de cada uno o en el botón de Edit. Una vez tengamos todos los valores debemos utilizar el filtro “MathExpression”. En expresión debemos introducir los mínimos y máximos tal como aparece en 2.5, en “ignoreRange” se indicará el índice del atributo, y en “invertSelection” lo seleccionaremos en true para que solo realice la normalización del atributo indicado. Este proceso debe realizarse para cada uno de los atributos, indicando para cada uno sus correspondientes valores mínimos y máximos.

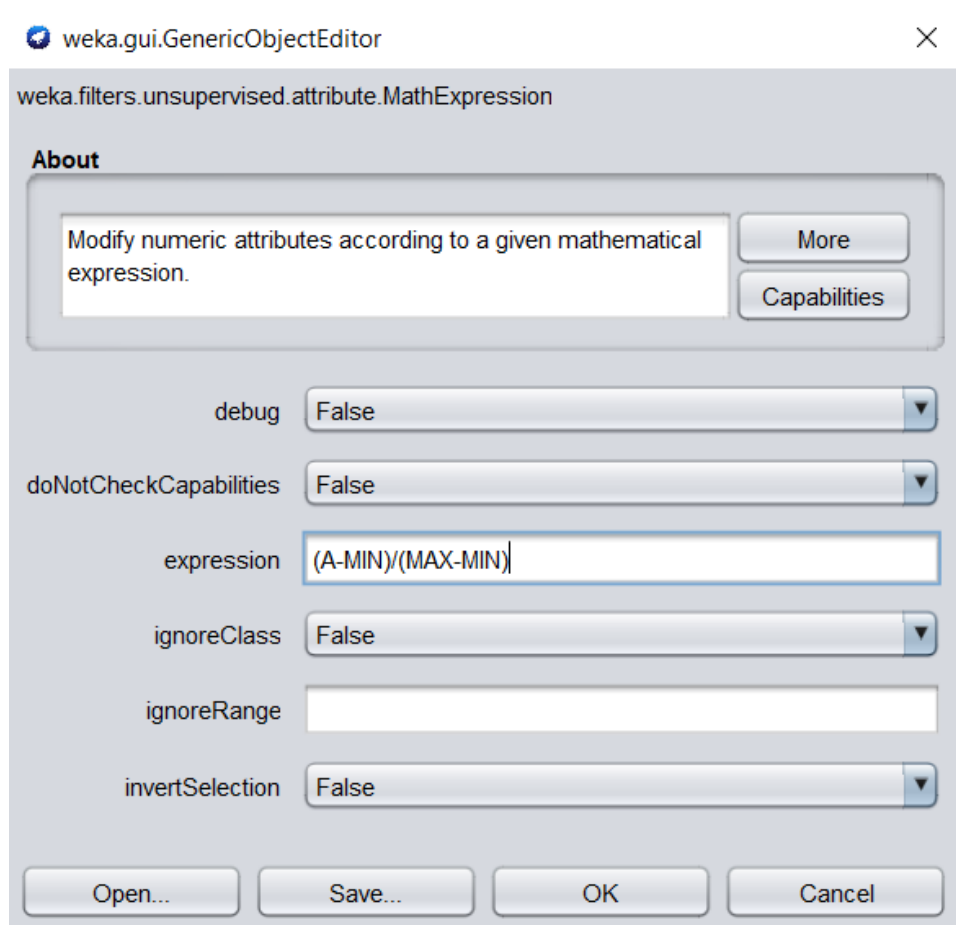


Fig. 2.5: Filtro Normalizacion



## Capítulo 3

# Entregable 3

### 1. Base de datos WINE

Escoja una de las bases de datos de la clasificación para el trabajo de las dispuestas en Moodle. Aplique el preprocesamiento adicional (si se puede aplicar) sobre: 1) reemplazamiento de datos perdidos, 2) normalización y 3) paso de nominal a binario u ordinal a numérico. Explique el procesamiento que haya llevado a cabo en los aspectos citados, y de no tener que hacerlo explique también el por qué.

Nuestro ejemplo elegido en este caso que ha sido wine.arff solo hemos tenido que normalizar los datos ya que no teníamos ningún dato perdido y además todos los atributos de nuestro fichero eran numéricos por lo que nos ha sido bastante sencillo.

Relation: wine-weka.filters.unsupervised.attribute Reorder-R1-13,14-weka.filters.unsupervised.attribute Reorder-R1-13,14-weka.filters.unsupervised.attribute Reorder-R1-13,14-weka.filters.unsupervised.attribute R													
No.	1: alcohol	2: acido_malico	3: ceninaz	4: alcalinidad_cenizas	5: magnesio	6: fenoles_totales	7: flavonoides	8: fenoles_no_flavonoides	9: proantocianinas	10: intensidad_color	11: tono	12: OD280	13: class
	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Nominal
1	14.23	1.71	2.43	15.6	127.0	2.8	3.06	0.28	2.29	5.64	1.04	3.92	1
2	13.2	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.4	1
3	13.16	2.36	2.67	18.6	101.0	2.8	3.24	0.3	2.81	5.68	1.03	3.17	1
4	14.37	1.95	2.5	16.8	113.0	3.85	3.49	0.24	2.18	7.8	0.86	3.45	1
5	13.24	2.59	2.87	21.0	118.0	2.8	2.69	0.39	1.82	4.32	1.04	2.93	1
6	14.2	1.76	2.45	15.2	112.0	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1
7	14.39	1.87	2.45	14.6	96.0	2.5	2.52	0.3	1.98	5.25	1.02	3.58	1
8	14.06	2.15	2.51	17.6	121.0	2.6	2.51	0.31	1.25	5.05	1.06	3.58	1
9	14.83	1.64	2.17	14.0	97.0	2.8	2.98	0.29	1.98	5.2	1.08	2.85	1
10	13.86	1.35	2.27	16.0	98.0	2.98	3.15	0.22	1.85	7.22	1.01	3.55	1
11	14.1	2.16	2.3	18.0	105.0	2.95	3.32	0.22	2.38	5.75	1.25	3.17	1
12	14.12	1.48	2.32	16.8	95.0	2.2	2.43	0.26	1.57	5.0	1.17	2.82	1
13	13.75	1.73	2.41	16.0	89.0	2.6	2.76	0.29	1.81	5.6	1.15	2.9	1

Fig. 3.1: Wine.arff antes de aplicar el filtro Normalize

## CAPÍTULO 3. ENTREGABLE 3

Como vemos en la figura anterior los valores originales de nuestro fichero están sin organizar por lo tanto aplicaremos el filtro para conseguir que todos queden entre cero y uno.

No.	1: alcohol	2: acido_malico	3: ceninaz	4: alcalinidad_cenizas	5: magnesio	6: fenoles_totales	7: flavonoides	8: fenoles_no_flavonoides	9: proantocianinas	10: intensidad_color	11: tono	12: OD280	13: clas
	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Numerico	Nomina
1	0.8421...	0.191699604...	0.5721...	0.25773195876288...	0.619565...	0.6275862068...	0.5738396...	0.28301886792452835	0.59305993690...	0.372013651877...	0.45...	0.97069...	1
2	0.5710...	0.205533596...	0.4171...	0.03092783505154...	0.326086...	0.5758620689...	0.5105485...	0.24528301886792453	0.27444794952...	0.254505119453...	0.46...	0.78021...	1
3	0.5605...	0.320158102...	0.7005...	0.41237113402061...	0.336956...	0.6275862068...	0.6118143...	0.320754716981132	0.75709779179...	0.375426621160...	0.44...	0.69597...	1
4	0.8769...	0.239130434...	0.6096...	0.31958762886597...	0.467391...	0.9896551724...	0.6645569...	0.2075471698113207	0.55835962145...	0.556313993174...	0.30...	0.79853...	1
5	0.5815...	0.365612648...	0.8074...	0.53608247422680...	0.521739...	0.6275862068...	0.4957805...	0.49056603773584906	0.44479495268...	0.25938565529...	0.45...	0.60805...	1
6	0.8342...	0.201581027...	0.5828...	0.23711340206185...	0.456521...	0.7896551724...	0.6434599...	0.39622641509433965	0.49211356466...	0.466723549488...	0.46...	0.57875...	1
7	0.8842...	0.223320158...	0.5828...	0.20618556701030...	0.282608...	0.5241379310...	0.4599156...	0.320754716981132	0.49526813880...	0.338737201365...	0.43...	0.84615...	1
8	0.7973...	0.278656126...	0.6684...	0.36082474226804...	0.554347...	0.5586206896...	0.4578059...	0.33962264150943384	0.26498422712...	0.321672354948...	0.47...	0.84615...	1
9	1.0	0.177865512...	0.4331...	0.17525773195876...	0.293478...	0.6275862068...	0.5586206...	0.3018867924528301	0.49526813880...	0.334470989761...	0.48...	0.57875...	1
10	0.7447...	0.120553359...	0.4886...	0.27835051546391...	0.304347...	0.6896551724...	0.5928270...	0.16981132075471697	0.45425887507...	0.508825938566...	0.43...	0.83516...	1
11	0.8078...	0.280612411...	0.5026...	0.38144329846607...	0.380474...	0.6793103448...	0.6286419...	0.16981132075471697	0.62145110410...	0.381389317406...	0.42...	0.69597...	1

Fig. 3.2: Wine.arff una vez normalizado

### 1.1. Algoritmo clasificación KNN

Con la base de datos escogida anteriormente, use el algoritmo de clasificación KNN con un 10-fold crossvalidation. Use un valor de vecinos  $k=3$  dejando por defecto el resto de parámetros.

Para realizar este ejercicio una vez ya tenemos preprocesada nuestra base de datos, nos vamos al apartado classify dentro de Weka, una vez aquí escogemos nuestro clasificador deseado, en este caso vamos a utilizar el clasificador KNN, conocido como algoritmo del vecino más cercano, para ello debemos de entrar dentro de la carpeta lazy en el cual se encuentra y seleccionar el clasificador ibk que es el nombre asignado en Weka para nuestro algoritmo. Una vez seleccionado debemos de modificar algunos valores por defecto ya que por defecto el numero de vecinos con el que comparar en uno es decir  $k=1$ , para nuestro problema en concreto tal y como nos indica en el enunciado debemos de cambiarlo y usar  $k=3$ , y además utilizar crossvalidation con 10 folds tal y como se muestra en la figura de abajo.

## 1. BASE DE DATOS WINE

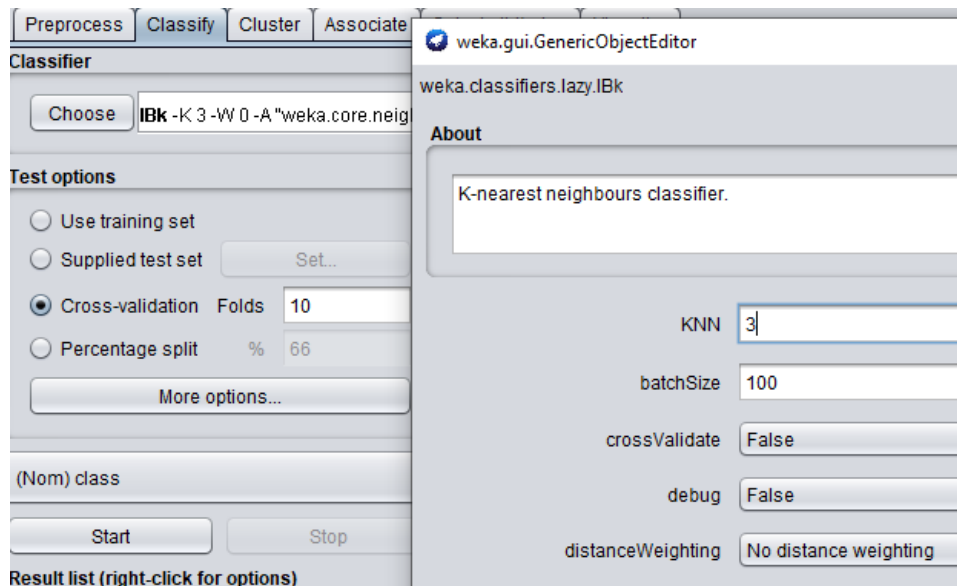


Fig. 3.3: Valores modificados para IBK, k=3

Una vez aplicado el clasificador sobre nuestra base de datos vemos como han sido los resultados:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      168           94.382 %
Incorrectly Classified Instances    10           5.618 %
Kappa statistic                    0.9153
Mean absolute error                 0.0586
Root mean squared error            0.1789
Relative absolute error            13.3445 %
Root relative squared error        38.1816 %
Total Number of Instances         178
```

Fig. 3.4: Estadísticas globales de nuestro clasificador

Nuestro clasificador ha realizado una muy buena clasificación obteniendo un total de 168 instancias bien clasificadas de las 178 totales con solo 10 erróneas. Ha conseguido un 94.382% de instancias bien clasificadas, frente a un 5.618% de instancias mal clasificadas.

### CAPÍTULO 3. ENTREGABLE 3

---

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,059	0,894	1,000	0,944	0,917	0,994	0,979	1
	0,859	0,000	1,000	0,859	0,924	0,886	0,978	0,969	2
	1,000	0,023	0,941	1,000	0,970	0,959	0,993	0,964	3
Weighted Avg.	0,944	0,026	0,949	0,944	0,943	0,916	0,987	0,971	

Fig. 3.5: Estadísticas de clasificación de cada clase

En la figura de arriba vemos una tabla en la cual nos da datos referentes a la clasificación más detallada de cada clase, cabe destacar los valores de TP Rate de la clase uno y dos(valores de uno para ambas clases) en las cuales todas las instancias han sido clasificadas de manera correcta, además debemos de tener en cuenta la media de ROC Área que es muy cercana a uno la cual nos indica que nuestro clasificador está muy bien entrenado.

En la siguiente figura, la matriz de confusión, nos indica como se han clasificado las instancias en cada clase, vemos como lógicamente coincide con lo comentado anteriormente y en la clase a(1) y c(3) todas las instancias han sido clasificadas de manera correcta, mientras que en la clase b(2) de las 71 instancias pertenecientes a esta clase ha sido bien clasificadas 61, siendo de estas 7 clasificadas de manera errónea en la clase a(1) y 3 en la clase c(3).

```
=== Confusion Matrix ===

  a   b   c   <-- classified as
59   0   0 |   a = 1
 7  61   3 |   b = 2
 0   0  48 |   c = 3
```

Fig. 3.6: Estadísticas de clasificación de cada clase

## 1.2. Algoritmo Simple Logistic

Con la base de datos escogida anteriormente, ejecute el algoritmo Simple-Logistic con 10-fold crossvalidation.

```
Class 1 :  
-3.06 +  
[alcohol] * 4.76 +  
[ceninaz] * 2.54 +  
[alcalinidad_cenizas] * -9.66 +  
[flavonoides] * 4.53 +  
[OD280] * 3  
  
Class 2 :  
10.15 +  
[alcohol] * -9.34 +  
[acido_malico] * -1.43 +  
[ceninaz] * -4.66 +  
[alcalinidad_cenizas] * 1.66 +  
[magnesio] * -2.29 +  
[intensidad_color] * -7.23  
  
Class 3 :  
9.37 +  
[flavonoides] * -27.16 +  
[tono] * -7.43 +  
[OD280] * -7.48
```

Fig. 3.7: Modelo de Regresión Logística obtenido al aplicar SampleLogistic

Una vez cargada nuestra base de datos Wine en Weka y preprocesada correctamente entramos en el apartado Classify para aplicar el algoritmo de regresión SampleLogistic, para ello entramos dentro de la carpeta functions que se encuentra a su vez dentro de la carpeta clasifiers y seleccionamos nuestro algoritmo. En el resultado obtenido tal y como se indica en la figura de abajo nos aparece las distintas rectas de regresión obtenidas por el

modelo, a partir de estas rectas mediante la función softmax obtenemos la probabilidad de pertenencia de una instancia a una de las clases. Además observamos (Fig3.7) como hay algunos atributos que no aparecen en estas funciones por lo tanto no son de utilidad a la hora de clasificar una instancia, estos atributos son, fenolestotales, fenolesnoflavonoides, proantocianinas.

En las siguientes ecuaciones que indican las rectas de regresión de las distintas clases vemos los valores de beta los cuales indican la importancia de las variables. Todas siguen la siguiente ecuación:

A continuación se expone la recta de regresión de la clase uno donde vemos que las variables que mas influyen son alcalinidad\_cenizas y alcohol.

$$f_1(x, \theta) = -3.06 + Alcohol * 4.76 + ceninaz * 2.54 + alcalinidad\_cenizas * -9.66 + flavonoides * 4.53 + OD280 * 3$$

La siguiente ecuación 3.1 nos muestra la recta de regresión de la clase dos donde debemos destacar que las variables mas influyentes son intensidad\_color y alcohol.

$$f_1(X, \hat{\theta}) = \beta_0 + \sum_{i=1}^n \hat{\beta}_i x_i \quad (3.1)$$

$$f_2(x, \theta) = 10.15 + Alcohol * 9.34 + ceninaz * -4.66 + alcalinidad\_cenizas * 1.66 + acidomalico * -1.43 + magnesio * -2.29 + intensidad\_color * -7.23$$

Por último tenemos la ecuación de regresión de la clase tres en la que destaca la importancia de la variable flavonoides.

$$f_3(x, \theta) = 9.37 + Flavonoides * -27.16 + tono * -7.43 + OD280 * -7.48$$

Con respecto a las métricas se tiene un CCR 95.5056 de un frente 94.38 al de un que daba el algoritmo KNN. El estadístico kappa 0.932 es igual a que como en el caso anterior quiere decir que el modelo obtenido es muy superior que uno basado en el azar. Por otro lado los valores de TP rate por clase son 0.983, 0.901 y 1 que son bastante similares a los obtenidos anteriormente.

## 1. BASE DE DATOS WINE

---

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,983	0,042	0,921	0,983	0,951	0,926	0,995	0,989	1
0,901	0,009	0,985	0,901	0,941	0,907	0,993	0,990	2
1,000	0,015	0,960	1,000	0,980	0,972	1,000	1,000	3
0,955	0,022	0,957	0,955	0,955	0,931	0,995	0,992	

Fig. 3.8: Estadísticas de clasificación de cada clase

## Capítulo 4

# Entregable 4

### 0.1. Ejercicio 1

Escoja una de las bases de datos de clasificación para el trabajo de las dispuestas en Moodle. Se entiende que además de pasarla a formato .arff ya ha aplicado el preprocesamiento necesario en función del fichero "Pistas sobre los datasets con posible preprocesamiento a simple vista.pdf", en el caso que sea una de las bases de datos que lo requiera.

Cargue la base de datos y ejecute el algoritmo C4.5 usando un 75 % para entrenar y un 25 % para generalizar, con los parámetros por defecto. Analice y muestre el árbol obtenido con los parámetros por defecto: nodo principal, número de nodos u hojas, variables presentes y omitidas. Comente también los resultados de las métricas obtenidas.

Para comenzar con el ejercicio hemos escogido la base de datos proporcionada por Moodle que anteriormente hemos pasado a formato .arff, esta base de datos consta de doce atributos de tipo numérico por lo cual deberíamos de aplicarle una discretización para así pasarlos a nominal, sin embargo esta es una de las ventajas del algoritmo empleado que a diferencia del algoritmo ID3 este, el C4.5 discretiza automáticamente por lo que para aplicarlo solo hemos tenido que modificar el tanto por ciento que deseamos destinar



---

a entrenamiento tal y como muestra la figura de abajo.

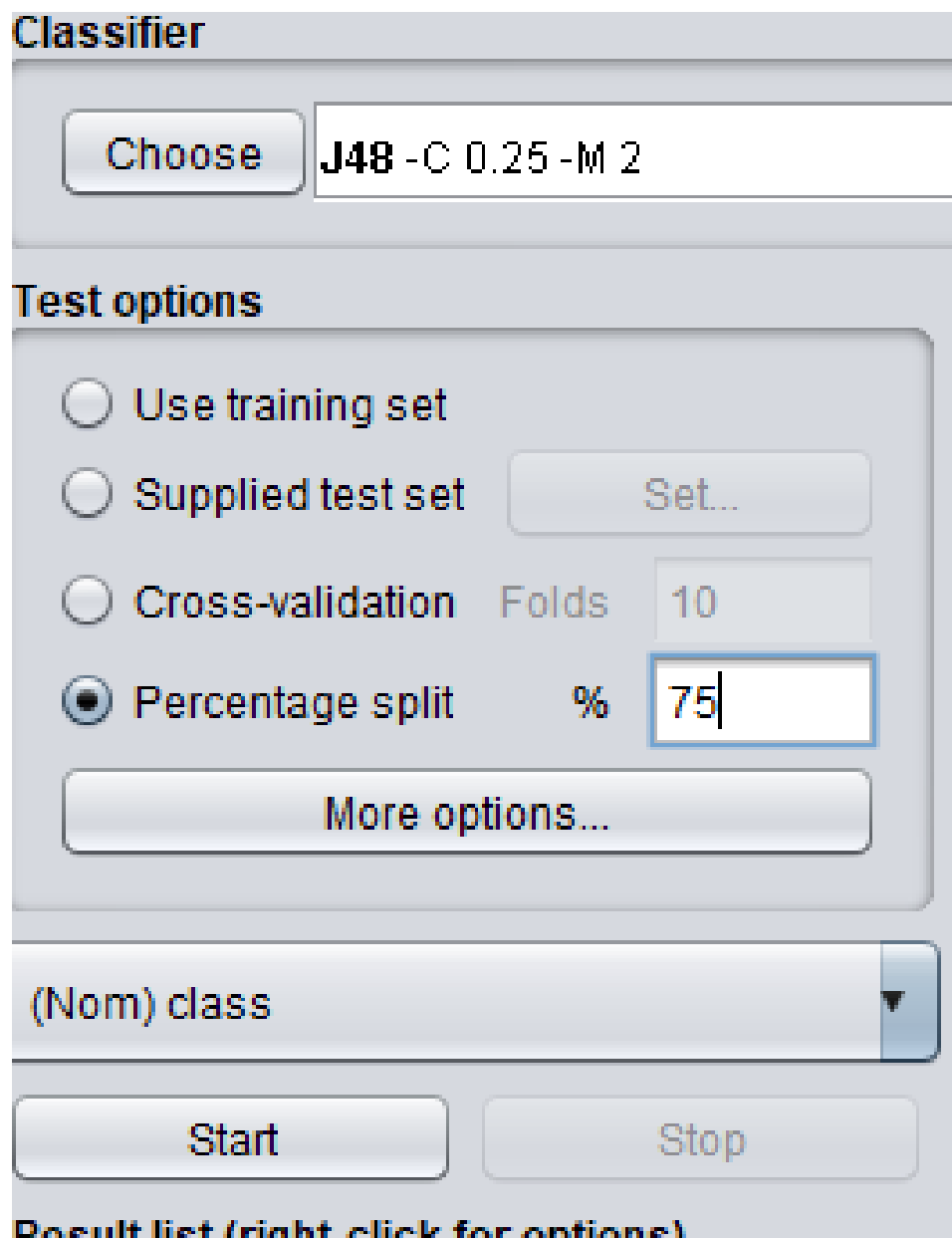


Fig. 4.1: Aplicación del 75 % de entrenamiento

A continuación mostraremos el árbol obtenido para comentar sus resultados así como sus reglas pertinentes y algunos datos de su estructura.

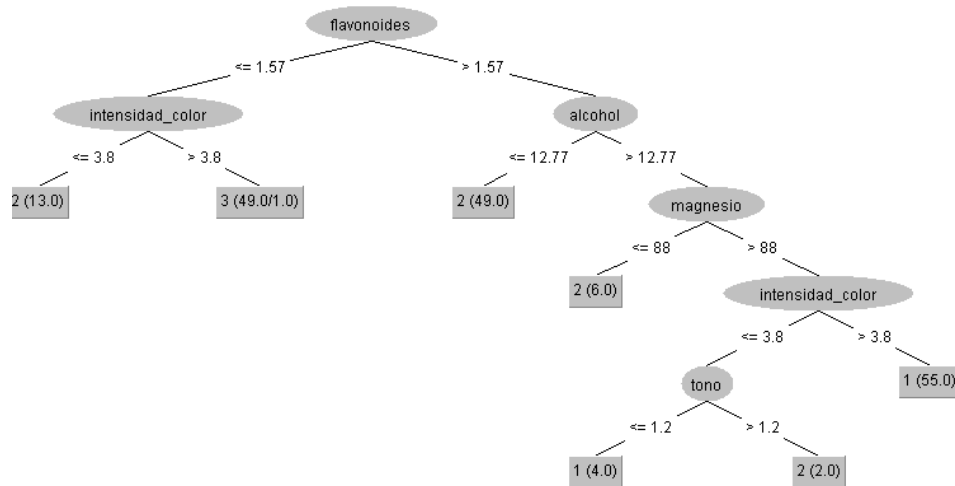


Fig. 4.2: Árbol obtenido por el algoritmo C4.5

```
flavonoides <= 1.57
|   intensidad_color <= 3.8: 2 (13.0)
|   intensidad_color > 3.8: 3 (49.0/1.0)
flavonoides > 1.57
|   alcohol <= 12.77: 2 (49.0)
|   alcohol > 12.77
|   |   magnesio <= 88: 2 (6.0)
|   |   magnesio > 88
|   |   |   intensidad_color <= 3.8
|   |   |   |   tono <= 1.2: 1 (4.0)
|   |   |   |   tono > 1.2: 2 (2.0)
|   |   |   intensidad_color > 3.8: 1 (55.0)
```

```
Number of Leaves   :      7
```

```
Size of the tree   :      13
```

Fig. 4.3: Reglas propias del árbol

---

Cómo vemos en las figuras de arriba nuestro árbol está formado por seis nodos que corresponde con seis análisis de variable que son, flavonoides, intensidad\_color, alcohol, magnesio, y tono. Comenzamos analizando flavonoide ya que es esta la que nos produce una menor entropía. Además poseemos siete nodos terminales u hoja los cuales determinan a que clase pertenece la instancia analizada según el camino o reglas seguidas. También podemos destacar que hay alguna variables las cuales no aparecen lo que nos quiere decir que no son influyentes a la hora de clasificar una instancia, estas son, ácido\_malico, ceninaz, alcalinidad\_cenizas, fenoles\_totales, fenoles\_no\_flavonoides, proantocianinas y OD280.

Con respecto a las métricas debemos destacar un alto valor de CCR con un 93.1818% además de un valor del estadístico Kappa de 0.8966 muy superior al 0.5 que representaría el azar. Con respecto a las métricas más específicas de cada clase debemos destacar un alto valor de TP Rate con valores de 0.882, 0.909 y 1 para cada clase respectivamente.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,882	0,000	1,000	0,882	0,938	0,906	0,941	0,928	1
0,909	0,061	0,833	0,909	0,870	0,825	0,904	0,780	2
1,000	0,036	0,941	1,000	0,970	0,953	0,982	0,941	3
0,932	0,028	0,937	0,932	0,932	0,903	0,947	0,896	

Fig. 4.4: Estadísticas de clasificación de cada clase

Con ayuda de la matriz de confusión proporcionada por Weka vemos como de diecisiete instancias a clasificar de la clase a(1) quince han sido clasificadas bien frente a dos que han sido clasificadas de manera errónea en la clase b(2), con respecto a la clase b(2) vemos cómo de once instancias, diez han sido bien clasificadas frente a una que ha sido clasificada como de clase c(3), por último destacar que en la clase c(3) todas han sido bien clasificada haciendo un total de dieciséis.

A continuación se muestra la matriz de confusión para una mejor comprensión de lo anterior.

**=== Confusion Matrix ===**

```
      a   b   c   <-- classified as
15    2    0 |   a = 1
 0   10    1 |   b = 2
 0    0   16 |   c = 3
```

Fig. 4.5: Estadísticas de clasificación de cada clase

**0.2. Ejercicio 2**

Escoja una de las bases de datos de clasificación para el trabajo de las dispuestas en Moodle. Se entiende que además de pasarla a formato .arff ya ha aplicado el preprocesamiento necesario en función del fichero "Pistas sobre los datasets con posible preprocesamiento a simple vista.pdf", en el caso que sea una de las bases de datos que lo requiera. Cargue la base de datos con un 75/25% y ejecute el algoritmo MultilayerPerceptron con los valores por defecto. ¿Qué observa al ir modificando solo el TrainingTime? ¿Cambia el valor de Correctly Classified instances al modificar el parámetro? ¿se estanca el aprendizaje o sobre entrena? ¿Qué observa al ir modificando solo el LearningRate? ¿Cambia el valor de Correctly Classified instances al modificar el parámetro? ¿se estanca el aprendizaje o sobre entrena?

Para realizar este ejercicio hemos escogido la base de datos wine.arff utilizada con anteriores ejemplos. De manera inicial con todos los valores por defecto y con un valor para TrainingTime de 500 tal y como viene por defecto el algoritmo nos clasifica 43 de las 44 destinadas a prueba de manera correcta, a medida que aumentamos el TrainingTime el valor del CCR sigue

---

constante por lo que el algoritmo se estanca, por tanto hemos optado por reducirlo a la unidad e ir incrementando hasta comprobar donde este algoritmo de estanca que ha sido en un valor de siete. En la figura de abajo se muestra con más claridad la evolución del CCR.

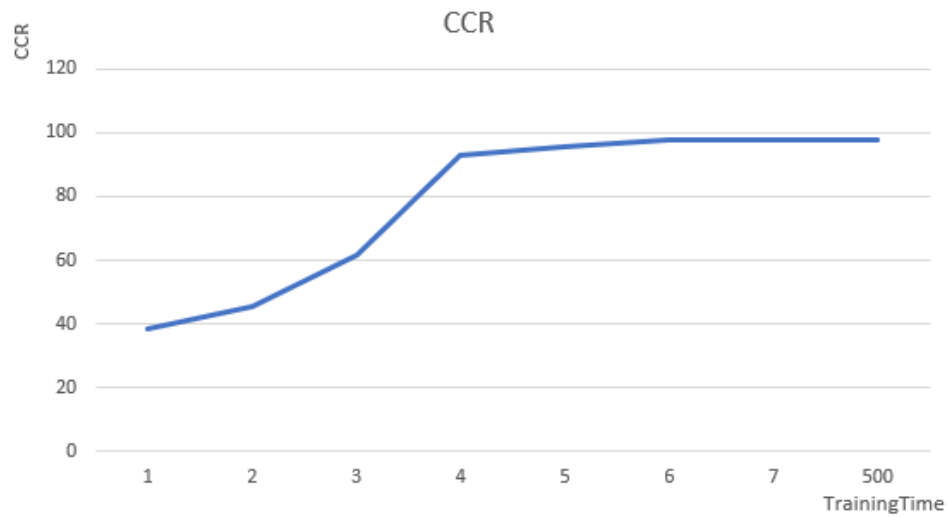


Fig. 4.6: Evolución del CCR con respecto a TrainingTime

# Bibliografía

- [1] *Introducción al aprendizaje automático*. 2019. <https://developers.google.com/machine-learning/crash-course/ml-intro?hl=es-419>.
- [2] *Introducción al aprendizaje automático*. 2019. <https://stackoverflow.com/search?q=machine+learning+weka>.
- [3] Departamento de informática y Análisis numérico. *Introducción al aprendizaje automático y Weka*. 2019. [https://moodle.uco.es/m1920/pluginfile.php/161272/mod\\_resource/content/23/practica1-1-.pdf](https://moodle.uco.es/m1920/pluginfile.php/161272/mod_resource/content/23/practica1-1-.pdf).
- [4] Departamento de informática y Análisis numérico. *Preprocesamiento y más filtros de Weka*. 2019. [https://moodle.uco.es/m1920/pluginfile.php/161313/mod\\_resource/content/11/practica2-.pdf](https://moodle.uco.es/m1920/pluginfile.php/161313/mod_resource/content/11/practica2-.pdf).
- [5] Departamento de informática y Análisis numérico. *Árboles y redes Neuronales*. 2019. [https://moodle.uco.es/m1920/pluginfile.php/161304/mod\\_resource/content/10/practica4-.pdf](https://moodle.uco.es/m1920/pluginfile.php/161304/mod_resource/content/10/practica4-.pdf).
- [6] Departamento de informática y Análisis numérico. *Regrasión y clasificación con Weka*. 2019. [https://moodle.uco.es/m1920/pluginfile.php/161289/mod\\_resource/content/15/practica3-.pdf](https://moodle.uco.es/m1920/pluginfile.php/161289/mod_resource/content/15/practica3-.pdf).

## BIBLIOGRAFÍA

---

---