

# PRACTICAS INTRODUCCIÓN AL APRENDIZAJE AUTOMÁTICO

Prácticas 1,2,3 -> Dataset: Criaturas tenebrosas

Práctica 4 -> Competición de Kaggle



UNIVERSIDAD DE CÓRDOBA

Introducción al Aprendizaje Automático  
Especialidad: Computación  
3º Curso | 2º Cuatrimestre  
Grado en Ingeniería Informática  
Escuela Politécnica Superior de Córdoba  
Universidad de Córdoba

## Contenido

|        |   |    |
|--------|---|----|
| 1.     | <b>Práctica 1. Introducción a Weka .....</b>  | 1  |
| 1.1.   | Base de datos Iris .....  | 1  |
| 1.2.   | Base de datos Audiology .....   | 7  |
| 1.3.   | Base de datos Criaturas Tenebrosas - Particionamiento .....   | 8  |
| 1.4.   | Base de datos Criaturas Tenebrosas .....  | 10 |
| 1.5.   | Descripción de filtros no Supervisados .....  | 13 |
| 1.5.1. | filters/unsupervised/attribute/Normalize .....  | 13 |
| 1.5.2. | filters/unsupervised/attribute/ReplaceMissingValues .....   | 14 |
| 1.5.3. | filters/unsupervised/attribute/NominalToBinary.....   | 14 |
| 1.5.4. | filters/unsupervised/instance/RemoveDuplicates .....  | 15 |
| 1.5.5. | filters/unsupervised/instance/Resample.....   | 15 |
| 1.5.6. | filters/unsupervised/attribute/Remove.....  | 15 |
| 1.5.7. | filters/unsupervised/attribute/RemoveUseless .....  | 16 |
| 1.6.   | Descripción de filtros Supervisados .....   | 16 |
| 1.6.1. | filters/supervised/attribute/Discretize .....   | 16 |
| 1.6.2. | filters/supervised/attribute/NominalToBinary .....  | 16 |
| 1.6.3. | filters/supervised/instance/SpreadSubsample.....  | 18 |
| 1.6.4. | filters/supervised/instance/Resample .....  | 18 |
| 1.7.   | Actualización de la base de datos Criaturas Tenebrosas .....  | 19 |
| 2.     | <b>Práctica 2. Parte 1 – Regresión y Clasificación Weka .....</b>   | 21 |
| 2.1.   | Algoritmo IBK para k=2 en el entorno EXPLORER .....   | 21 |
| 2.2.   | Algoritmo IBK para k=3 en el entorno EXPLORER .....   | 26 |
| 2.3.   | Algoritmo IBK en el entorno EXPERIMENTER.....   | 30 |
| 2.4.   | Algoritmo Logistic en el Entorno EXPLORER .....   | 32 |
| 2.5.   | Algoritmo SimpleLogistic en el Entorno EXPLORER .....   | 35 |
| 3.     | <b>Practica 2. Parte 2 – Clustering con Weka .....</b>  | 37 |
| 3.1.   | Algoritmo K-means en base de datos Iris, UseTrainingSet. ....   | 37 |
| 3.2.   | Algoritmo K-means en base de datos Criaturas Tenebrosas, K=Numero de Clases ...                             | 39 |
| 3.3.   | Cuestiones sobre el algoritmo Kmeans.....   | 40 |
| 3.4.   | Algoritmo K-means en base de datos Criaturas Tenebrosas, Classes to clusters evaluation.....                | 40 |
| 3.5.   | Algoritmo K-means en base de datos Iris, Classes to clusters evaluation .....                               | 41 |
| 3.6.   | Algoritmo HierarchicalClusterer en base de datos Criaturas Tenebrosas, Classes to clusters evaluation ..... | 43 |

|        |   |    |
|--------|---|----|
| 4.     | <b>Práctica 3. Árboles y redes neuronales</b>                             | 47 |
| 4.1.   | Algoritmo C4.5 en base de datos Criaturas Tenebrosas .....                | 47 |
| 4.2.   | Algoritmo MultilayerPerceptron en base de datos Criaturas Tenebrosas..... | 51 |
| 5.     | <b>Practica 4. Competición de Kaggle</b> .....                            | 55 |
| 5.1.   | Tratamiento de datos perdidos.....  | 55 |
| 5.1.1. | Eliminación de atributos .....  | 56 |
| 5.1.2. | Recuperación de datos perdidos.....                                       | 58 |
| 5.1.3. | Establecer mismas unidades .....  | 60 |
| 5.1.4. | Eliminación de correlaciones.....   | 63 |
| 5.1.5. | Valores perdidos y outliers.....  | 63 |
| 5.1.6. | Normalización.....  | 64 |
| 5.1.7. | Selección de características.....   | 66 |
| 5.1.8. | Búsqueda del mejor algoritmo y parámetros .....                           | 66 |

## 1. Práctica 1. Introducción a Weka

### 1.1. Base de datos Iris

#### 1.1.1. ¿Cuántos atributos caracterizan la base datos?

Son 5 y son los siguientes: *sepallength*, *sepalwidth*, *petallength*, *petalwidth*, *class*.

#### 1.1.2. ¿Se trata de regresión o de clasificación?

Se trata de una clasificación ya que el objetivo es asignar una determinada clase dado un conjunto de atributos.

#### 1.1.3. ¿Cuál es el rango de valores de petalwidth? ¿Y su media? ¿Y su desviación típica?

- Rango de valores: [0.1 - 2.5]
- Media: 1.199
- Desviación típica: 0.763

- 1.1.4. Utilizando el entorno Weka Explorer->Visualize, determinar qué atributo permite discriminar linealmente entre la clase iris-setosa y las otras dos clases.

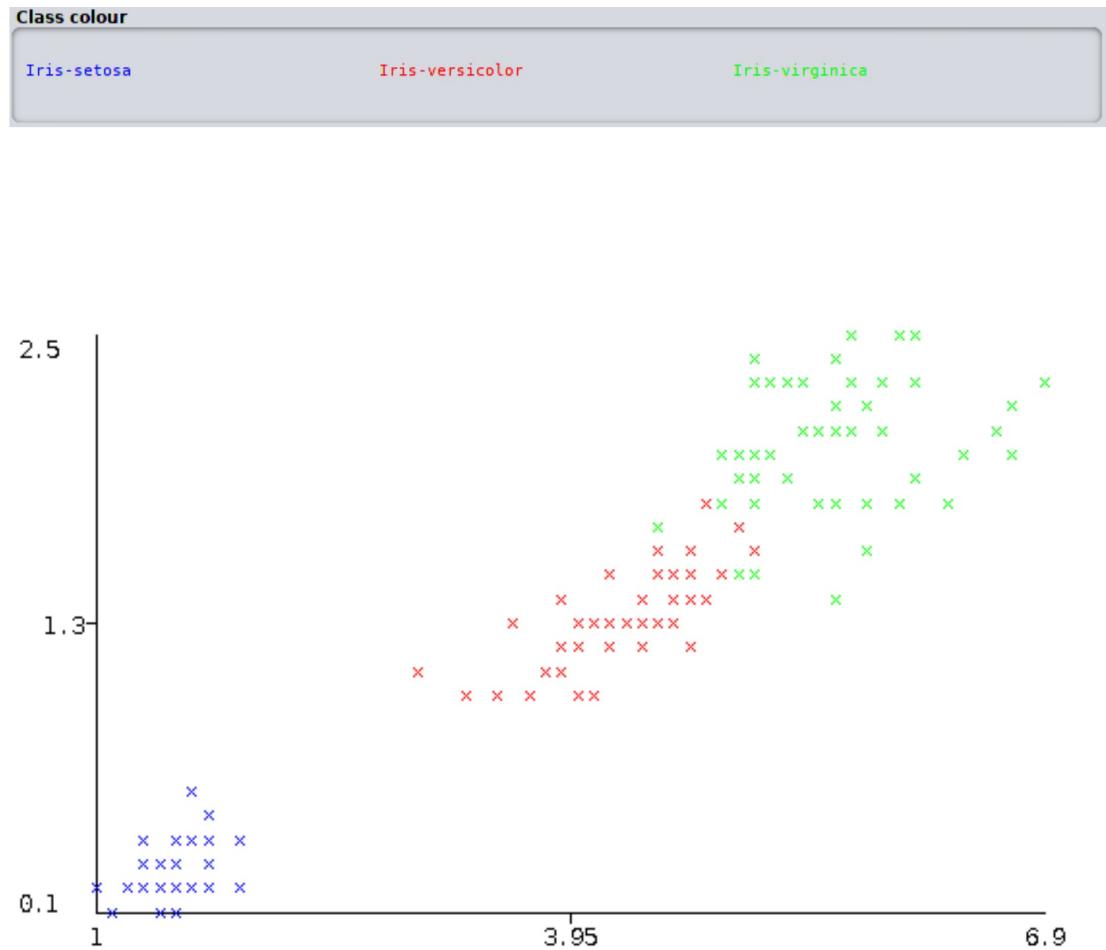


Figura 1.1: Base de datos Iris. Eje x: Petallength, eje y: Petalwidth

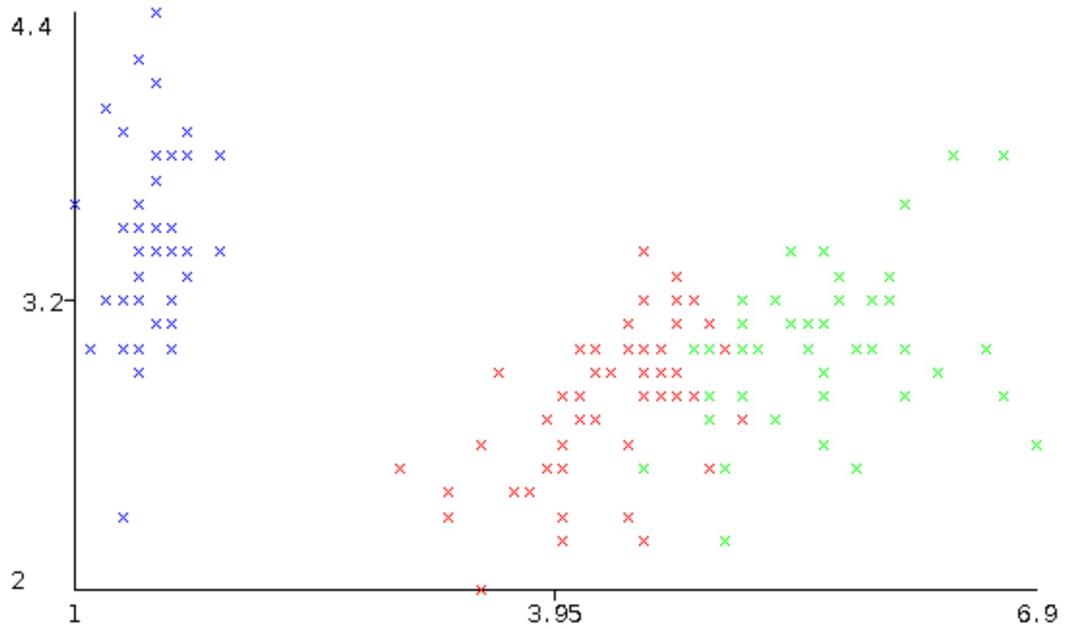


Figura 1.2: Base de datos iris. Eje x: Petallength, eje y: Sepalwidth

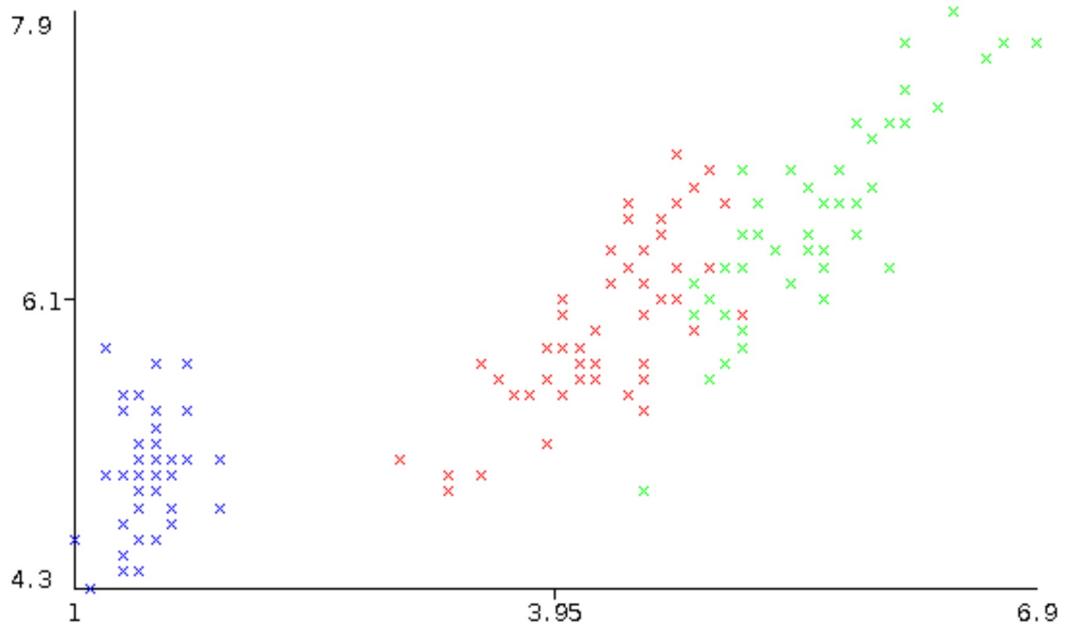


Figura 1.3: Base de datos iris. Eje x: Petallength, eje y: Sepallength

El atributo *Petallength*, siendo la clase *iris-setosa* el color azul, es el mejor atributo para discriminar linealmente la clase con respecto a las otras dos, tal y como se puede observar en las figuras 1.1, 1.2 y 1.3.

1.1.5. ¿Es posible separar linealmente la clase *iris-versicolor* de la clase *iris-virginica*?

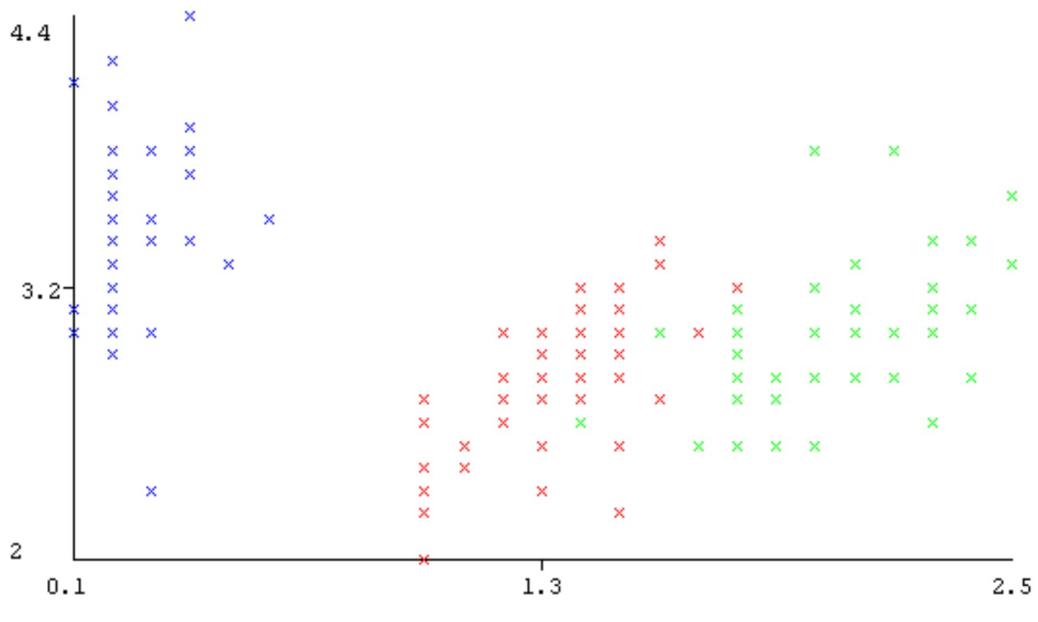


Figura 1.4: Base de datos Iris. Eje x: Petalwidth, eje y: Sepalwidth

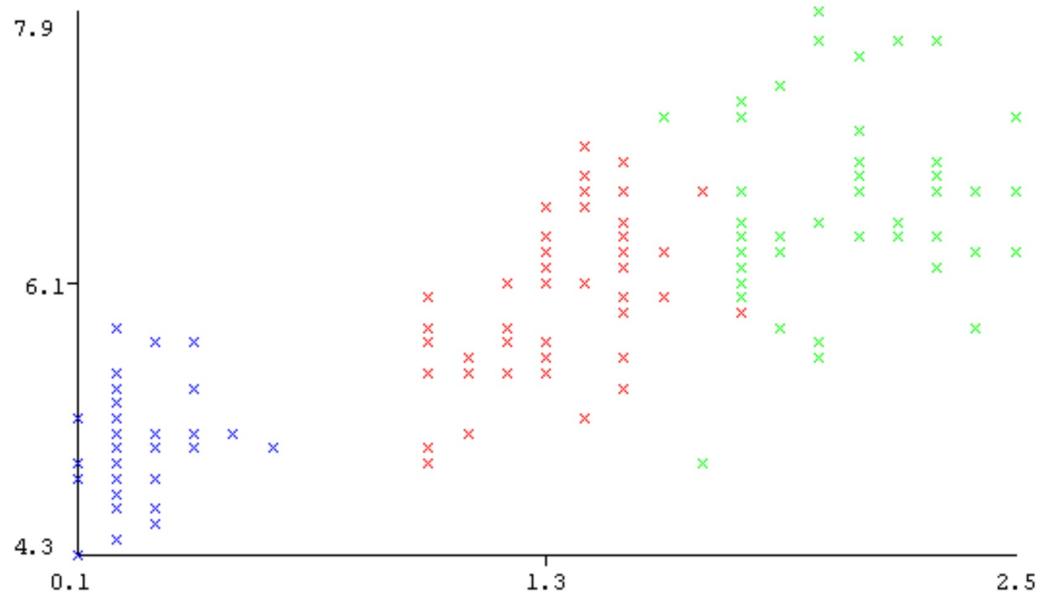


Figura 1.5: Base de datos Iris. Eje x: Petalwidth, eje y: Sepallength

Es posible separar la clase *Iris-versicolor* (color rojo) y la clase *Iris-virginica* (color verde) linealmente, sin embargo, existiría una probabilidad de fallo a la hora de clasificar, tal y como se puede observar en las *figuras 1.4 y 1.5*.

1.1.6. ¿Con qué dos atributos te quedarías para discriminar entre las tres clases del problema?

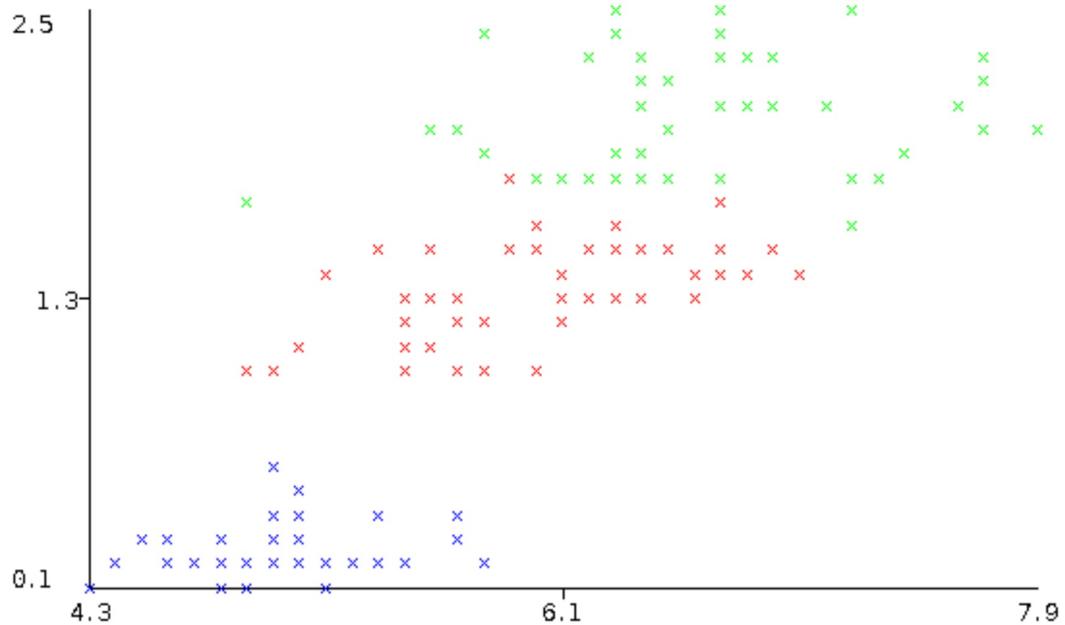


Figura 1.6: Base de datos Iris. Eje x: sepallength, eje y: Petalwidth

Utilizando los atributos *sepallength* y *petalwidth* se obtiene la mejor separación entre las distintas clases, sin que las instancias de una clase se mezclen con la de otra como se puede observar en la figura 1.6.

1.1.7. ¿Qué diferencia hay entre instancias *Distinct* y *Unique*? Fabríquese una base de datos propia para poner un ejemplo.

El número de instancias *Distinct* indica la cantidad de valores únicos distintos repetidos o no, mientras que el número de instancias *Unique* indica la cantidad de valores únicos no repetidos.

```

@relation prueba

@attribute Atributo1 numeric
@attribute Atributo2 numeric
@attribute Atributo3 numeric

@data
5,3,1
4,3,1
4,3,1
4,3,1
5,3,1
5,3,1|
4,3,1
5,3,1
4,2,1

```

*Figura 1.7: Base de datos de prueba*

En este caso, cada columna de datos se corresponde con un atributo.

|                 |               |
|-----------------|---------------|
| Name: Atributo1 | Type: Numeric |
| Missing: 0 (0%) | Distinct: 2   |

*Figura 1.8: Descripción de Atributo1*

Como se puede observar en la *figura 1.8*, el valor de *distinct* para el Atributo1 es igual a 2, esto se debe a que, como se puede observar en la figura 1.7, el Atributo1 toma únicamente los valores 5 y 4 mientras que el valor de *unique* es igual a 0 ya que el valor 5 y 4 es tomado por más de una instancia.

## 1.2. Base de datos Audiology

- 1.2.1. Aplique el filtro filters/unsupervised/attribute/NominalToBinary y describa cómo quedan ahora los atributos.

| No. | Name              |
|-----|-------------------|
| 1   | age_gt_60         |
| 2   | air               |
| 3   | airBoneGap        |
| 4   | ar_c              |
| 5   | ar_u              |
| 6   | bone              |
| 7   | boneAbnormal      |
| 8   | bser              |
| 9   | history_buzzing   |
| 10  | history_dizziness |
| 11  | class             |

Figura 1.9: Base de datos Audiology. Antes de aplicar el filtro NominalToBinary.

| No. | Name                |
|-----|---------------------|
| 1   | age_gt_60=t         |
| 2   | air=mild            |
| 3   | air=moderate        |
| 4   | air=normal          |
| 5   | air=profound        |
| 6   | air=severe          |
| 7   | airBoneGap=t        |
| 8   | ar_c=absent         |
| 9   | ar_c=elevated       |
| 10  | ar_c=normal         |
| 11  | ar_u=absent         |
| 12  | ar_u=elevated       |
| 13  | ar_u=normal         |
| 14  | bone=mild           |
| 15  | bone=moderate       |
| 16  | bone=normal         |
| 17  | bone=unmeasured     |
| 18  | boneAbnormal=t      |
| 19  | bser=normal         |
| 20  | history_buzzing=t   |
| 21  | history_dizziness=t |
| 22  | class               |

Figura 1.10: Base de datos Audiology. Tras aplicar el filtro NominalToBinary.

Tras aplicar el filtro el número de atributos pasa de ser once a ser veintidós. Cada atributo nominal se divide en varios atributos binarios que comienzan por el nombre del atributo original añadiendo el símbolo “=” más uno de los valores nominales, salvo en el caso de los atributos que sólo puedan tomar dos valores nominales, en cuyo caso, se crearía solo un nuevo atributo binario, identificando con el cero uno de los dos valores y con el uno el otro. Por ejemplo, el atributo nominal “air” puede tomar como valores *mild*, *moderate*, *normal*, *profound*, *severe*; por lo tanto, tras aplicar el filtro, surgirán como nuevos atributos: *air=mild*, *air=moderate*, *air=normal*, .... Este cambio se puede ver reflejado en la transición entre la *figura 1.9* y la *figura 1.10*.

#### 1.2.2. ¿Podría saber con antelación el número de atributos finales al aplicar este filtro?

Sí, dado que, como se ha explicado antes, los atributos con un número “n” superior a dos de posibles valores, se transforman en “n” nuevos atributos, mientras que los que sólo poseen dos posibles valores, se transforman en un único atributo.

### 1.3. Particione su base de datos Criaturas Tenebrosas usando filters/supervised/instance/StratifiedRemoveFolds

#### 1.3.1. 3.1. Divida el dataset en train y test mediante un numFolds=3. Describa el proceso realizado.

El proceso consiste en dividir el *dataset* en un determinado número de *folds* (pedazos, pliegues), en este caso tres, y que, al dividirlos, mantengan el orden del *dataset* inicial, es decir, si en el *dataset* inicial existían tres instancias de una supuesta clase A por cada dos instancias de una clase B, esta proporción ha de mantenerse también en cada uno de los *folds* en los que se divida el *datasets*, a esto se le denomina estratificación.

Concretamente, se exige un *3-fold*, es decir, dividir el *dataset* en tres partes (*folds*) y utilizar el *fold 1* para generalizar y el resto para entrenar, luego el *fold 2* para generalizar y el resto para entrenar...

Para poder realizar este procedimiento mediante la aplicación del filtro *StratifiedRemoveFolds*, se seguirán los siguientes pasos:

Primero, se aplica el filtro con los siguientes parámetros (*Figura 1.11*):

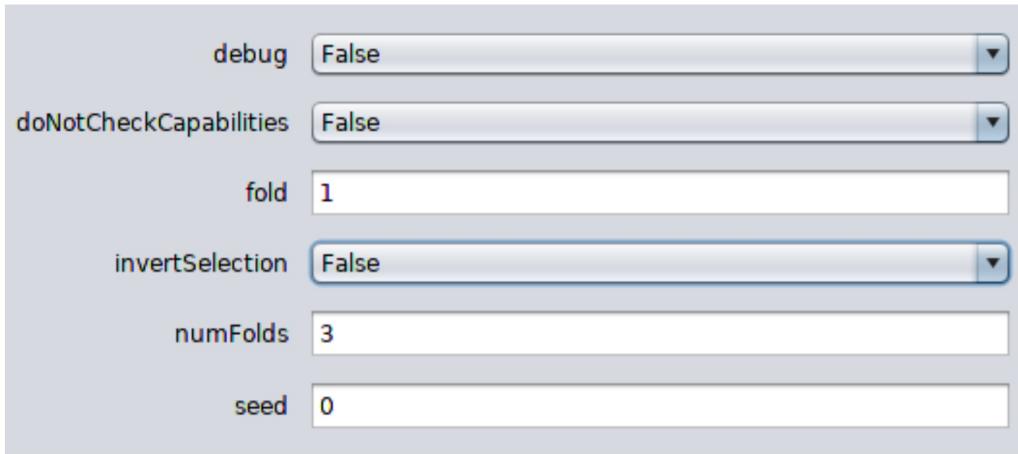


Figura 1.11: Parámetros para el filtro *StratifiedRemoveFolds*.

El número de *fold* en el cual se dividirá el *dataset*, se indicará en el parámetro *numFolds*, en este caso tres.

Posteriormente, se alternará el parámetro *invertSelection*. Cuando a este se le asigne el valor *false* para aplicar el filtro, se seleccionará el *fold* indicado en el parámetro *fold* (partición para *test*), mientras que cuando se le asigne *true*, se tomarán el resto de particiones (particiones para *train*), diferentes al *fold* seleccionado. Cada vez que se aplique el filtro se tendrá que almacenar el *dataset* resultante.

Por tanto, se ejecutará el filtro alternando los parámetros *fold*, entre 1 y 3, e *invertSelection*, entre *false* y *true*, manteniendo el parámetro *numFolds* a 3, con el fin de obtener todas las combinaciones posibles de *train* y *test* para 3-folds.

### 1.3.2. Divida el dataset en train y test mediante un *holdOut* con un 75% train y 25 % test. Describa el proceso realizado.

Se realizará un procedimiento similar al del apartado anterior, no obstante, variando el *numFolds* por un valor igual de 4. De esta manera, la partición destinada al *test* será de un 25%. Así, se podrá seleccionar cualquiera de los 4 *folds* resultantes como conjunto de *test*.

Otra alternativa posible es el *k-holdOut*, en el cual se mantendrá siempre el mismo *fold* seleccionado, como parámetro, para generalización (*test*), sin embargo, se modificará el parámetro *seed* para que los patrones seleccionados sean distintos en cada aplicación del filtro (garantiza la estratificación).

## 1.4. Base de datos de Criaturas Tenebrosas

- 1.4.1. Construya a partir del fichero dataset371.csv un fichero.arff para Weka. Ponga nombres de atributos descriptivos y use las herramientas que considere necesarias. Estudie previamente cual es el formato de Weka para un dataset.

El formato que utiliza *Weka* para sus *datasets* contiene un encabezado que proporciona metadatos acerca del tipo de dato de las columnas. Las distintas directivas de este encabezado comienzan con el símbolo (@). Hay una directiva para indicar el nombre del *dataset* (@RELATION), otra para indicar el nombre y el tipo de los distintos atributos (@ATTRIBUTE) y una para indicar el comienzo de los datos (@DATA). Los datos van separados por comas.

Para obtener un *fichero.arff* a partir de un fichero .csv tenemos 3 alternativas:

- Crearlo de forma manual.
- Cargar el archivo .csv en la herramienta *ArffViewer* de *Weka* y guardarla con formato *ARFF*.
- Cargar el archivo .csv directamente en *Weka Explorer* y guardarla con formato *ARFF*.

En este caso se va a emplear la herramienta *ArffViewer*, está se encuentra en el apartado *tools* de *Weka*. Una vez cargado el *dataset* en esta herramienta se puede ver cómo, automáticamente, ha establecido el nombre de los atributos del dataset.csv y también detecta automáticamente el tipo de los atributos. Lo único que habrá que hacer, por tanto, es guardar este fichero. Quedará un fichero como el siguiente:

```
@relation dataset371

@attribute id numeric
@attribute bone_length numeric
@attribute rotting_flesh numeric
@attribute hair_length numeric
@attribute has_soul numeric
@attribute color {clear,green,black,white,blue,blood}
@attribute type {Ghoul,Goblin,Ghost}

@data
0,0.354512,0.350839,0.465761,0.781142,clear,Ghoul
1,0.57556,0.425868,0.531401,0.439899,green,Goblin
2,0.467875,0.35433,0.811616,0.791225,black,Ghoul
3,0.776652,0.508723,0.636766,0.884464,black,Ghoul
4,0.566117,0.875862,0.418594,0.636438,green,Ghost
5,0.40569,0.252277,0.44142,0.380224,green,Goblin
```

Figura 1.12: Dataset criaturas misteriosas en formato ARFF.

1.4.2. Abra el entorno de Weka Explorer -> Preprocess de Weka, cargue la base de datos y describa de forma concienzuda tanto los atributos como las clases.

- **id**: identificador para las 371 criaturas detectadas.
- **bone\_length**: indica la longitud media de los huesos de la criatura. Está normalizado en el intervalo [0-1]. No hay dos criaturas que comparten la misma longitud media de los huesos ya que el valor de *unique* es 371 que coincide con el número de instancias. Los valores de las criaturas detectadas van desde [0.061-0.817] con una media y desviación típica igual a 0.434 y 0.133 respectivamente.
- **rotting\_flesh**: porcentaje de carne podrida en la criatura entre 0 y 1. No hay dos criaturas con el mismo porcentaje ya que el valor de *unique* es igual al número de instancias. Los valores de las criaturas detectadas van desde [0.096-0.932] con una media y desviación típica igual a 0.507 y 0.146 respectivamente.
- **hair\_length**: longitud media del pelo, normalizada entre 0 y 1. Hay dos criaturas que comparten la misma longitud media del pelo ya que el valor de *distinct* es una unidad menos que el número de instancias y por tanto *unique* es dos unidades menos que el número de instancias. El rango de valores de las criaturas detectadas es [0.135-1] con una media y desviación típica igual a 0.529 y 0.17 respectivamente.
- **has\_soul**: porcentaje de alma de la criatura entre 0 y 1. No hay dos criaturas con el mismo porcentaje de alma ya que el valor de *unique* es igual al número de instancias. Los valores de las criaturas detectadas están en el intervalo [0.009-0.936] con una media y desviación típica igual a 0.471 y 0.176 respectivamente.
- **color**: color dominante de la criatura. Que puede tomar los valores *clear*, *green*, *black*, *white*, *blue* y *blood* haciendo 120, 42, 41, 137, 19 y 12 criaturas de cada tipo respectivamente.
- **Clases**: Los distintos tipos de clases que hay son: *Ghoul*, *Goblin* y *Ghost* habiendo de cada una 129, 125 y 117 instancias respectivamente.

1.4.3. Observe si hay atributos identificadores. Si no existen diga por qué.

El atributo identificador es el *id*, que no aporta más información que la de indicar que cada instancia es única, lo cual es necesario para un buen aprendizaje del *dataset*, pero no aporta información sobre la clase.

1.4.4. Use el entorno Visualice, ¿hay alguna relación que sea visualmente significativa?

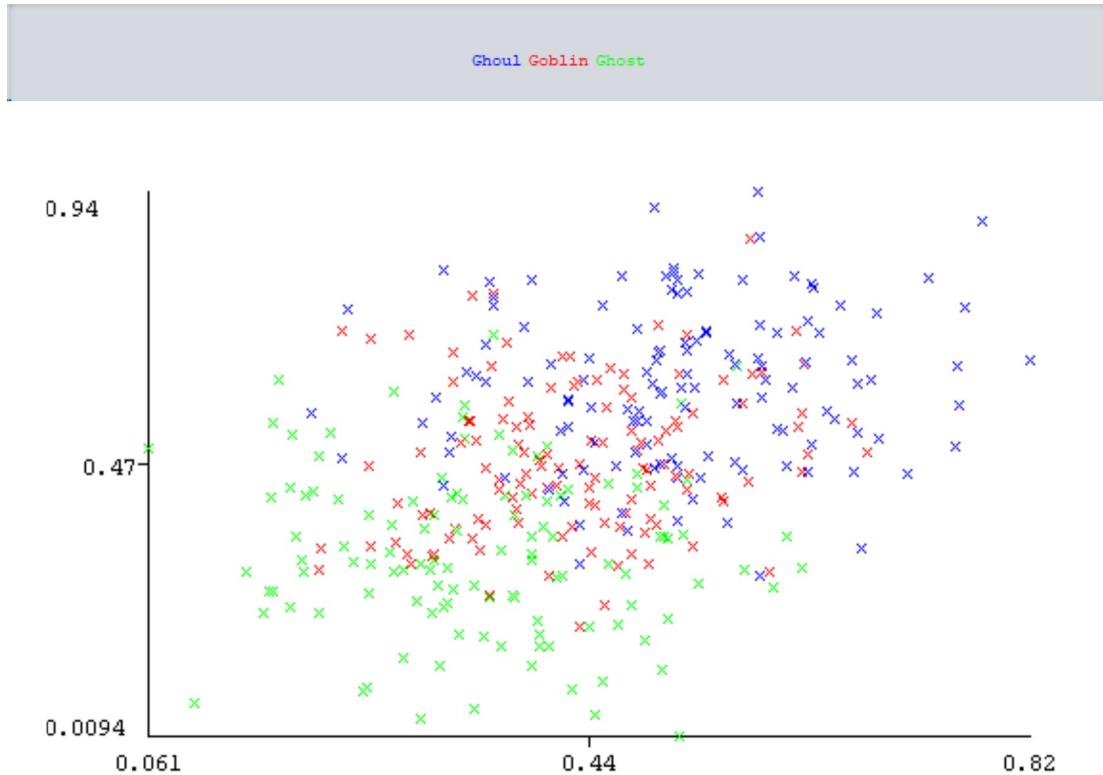


Figura 1.13:Base de datos Criaturas misteriosas. Eje x: bone\_length, eje y: has\_soul

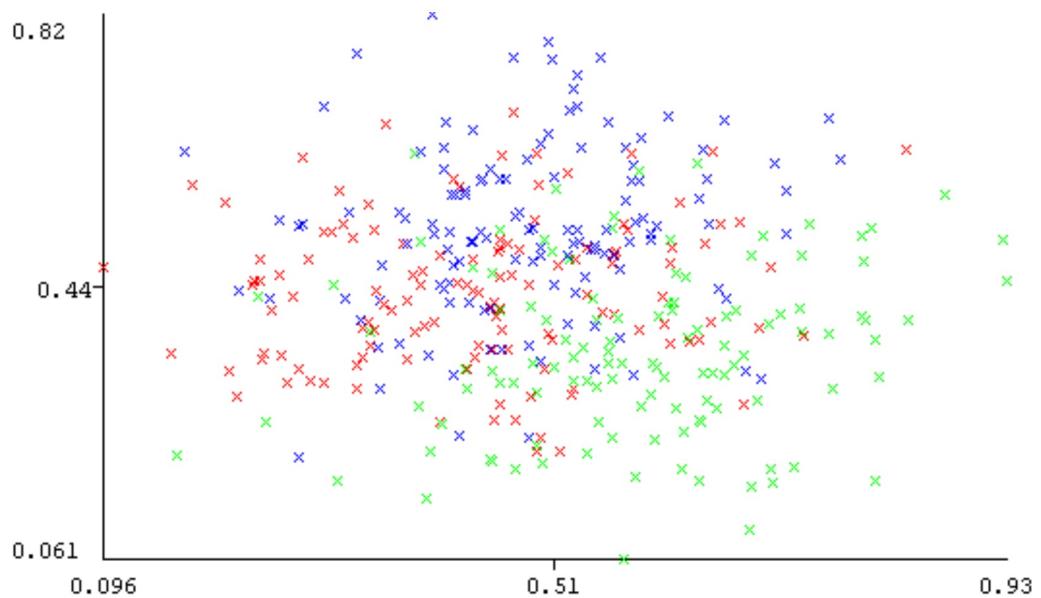


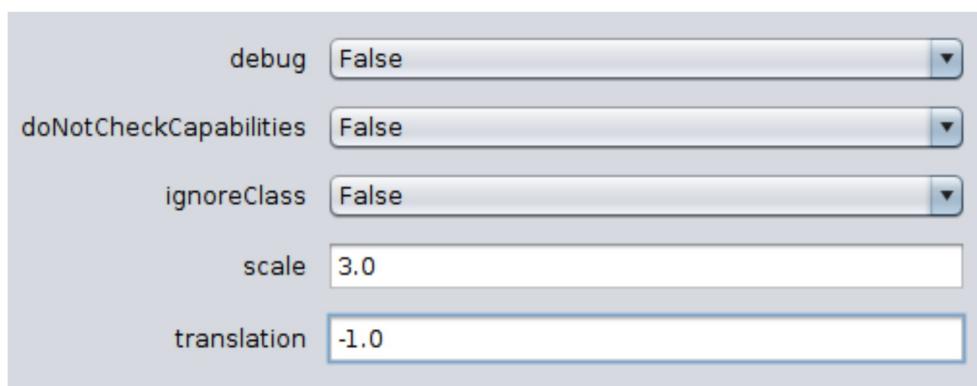
Figura 1.14:Base de datos Criaturas misteriosas. Eje x: rotting\_flesh, eje y: bone\_length

Como se aprecia en la *figura 1.13* y en la *figura 1.14*, existe una tendencia a que los valores de los atributos de la clase *goblin* (*roja*), se sitúen entre los valores de los atributos de la clase *ghoul* (*azul*) y *ghost* (*verde*).

## 1.5. Descripción de filtros no Supervisados

### 1.5.1. filters/unsupervised/attribute/Normalize

El filtro consiste en normalizar los atributos de tipo numérico a un especificado rango de valores, por defecto el rango se encontrará entre 0 y 1.



*Figura 1.15: Parámetros de ejemplo para el filtro no supervisado normalize.*

Con el parámetro *scale* se indica el valor máximo que se puede tomar en el intervalo (el mínimo es cero) mientras que con el parámetro *translation* se puede modificar el intervalo de salida. Por ejemplo, con un valor de escala 3.0 el intervalo quedaría de la forma [0,3] que, al aplicarle *translation*, queda como [-1,2] tal y como se puede observar en la *figura 1.16*.

| Name:     | bone_length | Type:     | Numeric |
|-----------|-------------|-----------|---------|
| Missing:  | 0 (0%)      | Distinct: | 371     |
| Unique:   | 371 (100%)  |           |         |
| Statistic | Value       |           |         |
| Minimum   | -1          |           |         |
| Maximum   | 2           |           |         |
| Mean      | 0.481       |           |         |
| StdDev    | 0.527       |           |         |

*Figura 1.16: Resultados del filtro sobre el atributo bone\_length.*

### 1.5.2. filters/unsupervised/attribute/ReplaceMissingValues

El filtro consiste en reemplazar todos aquellos valores perdidos tanto nominales como numéricos, usando para ello modelos basados en las medias y modas de sus atributos correspondientes.

```
@data
0,0.354512,0.350839,0.465761,?,clear,Ghoul
1,0.57556,0.425868,?,0.439899,green,Goblin
2,0.467875,0.35433,0.811616,0.791225,black,Ghoul
3,0.776652,0.508723,0.636766,0.884464,black,Ghoul
4,0.566117,0.875862,0.418594,0.636438,green,Ghost
5,0.40568,0.253277,0.44142,0.280324,green,Goblin
```

Figura 1.17: Base de datos Criaturas. Dos datos modificados a perdidos.

Se han modificado dos valores de distintos atributos (*Figura 1.17*) y distintas criaturas con el fin de poner un ejemplo práctico del funcionamiento del filtro, estos valores se han transformado a perdidos mediante la inclusión de un símbolo de interrogación en su respectivo lugar.

<https://machinelearningmastery.com/load-csv-machine-learning-data-weka/>

```
@data
0,0.354512,0.350839,0.465761,0.470555,clear,Ghoul
1,0.57556,0.425868,0.529108,0.439899,green,Goblin
2,0.467875,0.35433,0.811616,0.791225,black,Ghoul
3,0.776652,0.508723,0.636766,0.884464,black,Ghoul
4,0.566117,0.875862,0.418594,0.636438,green,Ghost
5,0.40568,0.253277,0.44142,0.280324,green,Goblin
```

Figura 1.18: Base de datos Criaturas. Datos reemplazados mediante ReplaceMissingValues.

Tras aplicar el filtro, los valores perdidos se han reemplazado por aproximaciones a sus verdaderos valores, como se puede observar en la *figura 1.18*.

### 1.5.3. filters/unsupervised/attribute/NominalToBinary

Como se explicó en el *apartado 1.2.1*, este filtro se encarga de transformar los atributos que son nominales en binarios, por ejemplo, en el caso del atributo color el resultado de

aplicar el filtro será un nuevo atributo para cada uno de los posibles colores y, cada atributo tomará los valores 0 o 1 en función de si la instancia tiene ese color de pelo o no. Uno de los parámetros que tiene el filtro es *attributeIndices* con el cual se puede indicar sobre qué atributos nominales se desea aplicar el filtro. Tras aplicar el filtro, el único atributo que se ve modificado es “color” que se divide en seis atributos binarios, uno por cada uno de sus posibles valores nominales. Si el atributo clasificador es de tipo numérico, el resultado que se produce es el mismo.

Otro parámetro que presenta este filtro es *BinaryAttributeNominal* que si se encuentra a true los atributos binarios resultantes serán de tipo nominal, es decir, el 0 se representará con una f mientras que el 1 se representa con una t.

#### 1.5.4. filters/unsupervised/instance/RemoveDuplicates

El filtro consiste en analizar el *dataset* y borrar aquellas tuplas que se encuentren duplicadas. En el *dataset* de criaturas no produce ningún efecto, pues no hay ninguna fila duplicada, en el caso de añadir una, el filtro la eliminaría.

#### 1.5.5. filters/unsupervised/instance/Resample

El filtro consiste en obtener un subconjunto o un superconjunto de valores del *dataset* original. En ambos casos, el conjunto resultante se obtendría mediante muestreo. Si se trata del subconjunto, este muestreo puede realizarse con o sin reemplazamiento, es decir, si se escoge una instancia, si el muestreo es con reemplazamiento, esta se podría volver a seleccionar dando a lugar a instancias duplicadas, mientras que sin reemplazamiento no, esto se indica mediante el parámetro *noReplacement*. Si se trata del superconjunto, el muestreo se realizará con reemplazamiento, o sea, dando lugar a instancias repetidas. El parámetro *sampleSizePercent* es el que permite seleccionar si se desea un subconjunto o un superconjunto especificando un porcentaje relacionado con el conjunto original (un 100% sería el conjunto completo); si escoge un porcentaje mayor de 100 %, el resultado será un superconjunto, mientras que, si es menor, será un subconjunto.

#### 1.5.6. filters/unsupervised/attribute/Remove

Este filtro se utiliza para borrar un rango de atributos del *dataset*. Con el parámetro *attributeIndices* se puede indicar que atributos borrar separándolos por comas, así como un intervalo de atributos. Por ejemplo: 1-3, 4,7. En este caso se borrarán los atributos 1,2,3,4 y 7. Otro parámetro que posee el filtro es *invertSelection*, el cual se utiliza para indicar aquellos atributos que se desean conservar (indicados en *attributeIndices*) y que el resto sean eliminados.

### 1.5.7. filters/unsupervised/attribute/RemoveUseless

Este filtro consiste en analizar el *dataset* y eliminar aquellos atributos que no son representativos, dicho de otro modo, que los valores que toman las diferentes instancias no varían demasiado entre sí basándose en un porcentaje de varianza, de manera que, el propio filtro da la opción de modificar dicho porcentaje de varianza con el parámetro *maximumVariancePercentajeAllowed*.

En el caso de la base de datos de criaturas, tras aplicar el filtro con sus valores por defecto y con el parámetro *maximumVariancePercentajeAllowed* en 90, no cambia nada dado que los valores son lo suficientemente dispersos como para que este filtro no surta efecto.

## 1.6. Descripción de filtros Supervisados

### 1.6.1. filters/supervised/attribute/Discretize

Este filtro se encarga de discretizar una serie de atributos numéricos indicados en atributos nominales mediante etiquetas. Este proceso se realiza mediante la división del dominio real de los posibles valores cada atributo en intervalos, de manera que, a cada intervalo se le asigna una etiqueta.

| Name: bone_length |                      | Type: Nominal |                |
|-------------------|----------------------|---------------|----------------|
| Missing: 0 (0%)   |                      | Distinct: 4   | Unique: 0 (0%) |
| No.               | Label                | Count         | Weight         |
| 1                 | '(-inf-0.198912]'    | 16            | 16.0           |
| 2                 | '(0.198912-0.405...' | 142           | 142.0          |
| 3                 | '(0.405882-0.627...' | 187           | 187.0          |
| 4                 | '(0.627693-inf)'     | 26            | 26.0           |

Figura 1.19: Atributo *bone\_length* tras aplicar el filtro *Discretize*

Como se puede observar en la figura 1.19, el atributo *bone\_length*, tras aplicar el filtro, toma cuatro posibles valores, siendo cada uno un intervalo distinto.

### 1.6.2. filters/supervised/attribute/NominalToBinary

El filtro funciona de manera similar al filtro no supervisado (explicado con anterioridad), con la diferencia de que, si el atributo clase es de tipo numérico entonces los atributos nominales con “n” posibles valores se transformarán en “n-1” atributos binarios de la manera descrita en “Clasificación y árboles de regresión”.

Para ver esto se aplicará el filtro sobre el siguiente dataset (*Figura 1.20*):

```
@relation dataset371

@attribute id numeric
@attribute bone_length numeric
@attribute rotting_flesh numeric
@attribute hair_length numeric
@attribute has_soul numeric
@attribute color {clear,green,black,white,blue,blood}
@attribute type numeric

@data
0,0.354512,0.350839,0.465761,0.781142,clear,1
1,0.57556,0.425868,0.531401,0.439899,green,1
2,0.467875,0.35433,0.811616,0.791225,black,2
3,0.776652,0.508723,0.636766,0.884464,black,2
4,0.566117,0.875862,0.418594,0.636438,green,3
5,0.40568,0.253277,0.44142,0.280324,green,1
6,0.399331,0.568952,0.618391,0.467901,white,1
```

Figura 1.20: Pequeño dataset en el que el atributo *type* es de tipo número.

Tras aplicar el filtro, el atributo *color* dará lugar a cinco nuevos atributos, ya que el atributo *color* toma 6 posibles valores como se puede ver en la *figura 1.21*.

```
@relation dataset371-weka.filters.supervised.attribute.NominalToBinary

@attribute id numeric
@attribute bone_length numeric
@attribute rotting_flesh numeric
@attribute hair_length numeric
@attribute has_soul numeric
@attribute 'color=white,blood,blue,green,black' numeric
@attribute 'color=blood,blue,green,black' numeric
@attribute 'color=blue,green,black' numeric
@attribute 'color=green,black' numeric
@attribute color=black numeric
@attribute type numeric

@data
0,0.354512,0.350839,0.465761,0.781142,0,0,0,0,0,1
1,0.57556,0.425868,0.531401,0.439899,1,1,1,1,0,1
2,0.467875,0.35433,0.811616,0.791225,1,1,1,1,1,2
3,0.776652,0.508723,0.636766,0.884464,1,1,1,1,1,2
4,0.566117,0.875862,0.418594,0.636438,1,1,1,1,0,3
5,0.40568,0.253277,0.44142,0.280324,1,1,1,1,0,1
6,0.399331,0.568952,0.618391,0.467901,1,0,0,0,0,1
```

Figura 1.21: Resultado de aplicar el filtro supervisado *NominalToBinary* sobre un dataset con una clase de tipo numérica.

Para ver qué color tiene asociada cada tupla se sigue una clasificación en forma de árbol. Por ejemplo, si el valor de todos los atributos resultantes en una tupla es cero quiere decir que el *color* es *clear*. Cuando el pelo es verde valdrá uno los cuatro primeros atributos y cero el último. Es decir, en aquellos atributos donde se encuentre presente el valor del color de pelo tomarán el valor 1 y se seguirá profundizando en el árbol de decisión hasta que se

llegue a un nodo (atributo) en el que ya no se encuentra ese *color* y por tanto se finalice el proceso de clasificación.

### 1.6.3. filters/supervised/instance/SpreadSubsample

El filtro funciona de manera similar al filtro *Resample*, es decir, extrae una muestra del conjunto de datos, con la diferencia de que, en lugar de mantener las proporciones en la submuestra, permite seleccionar las proporciones que tendrán las clases en la nueva muestra. Tras aplicar el filtro modificando la opción *distributionSpread* a 1.0 (distribución uniforme), las clases contienen ahora todas exactamente la misma cantidad de instancias, en este caso, se reducen todas al tamaño de la clase que poseía menos instancias inicialmente (117). También proporciona la posibilidad de indicar el número máximo de instancias de cada clase que se quieren seleccionar utilizando el parámetro *maxCount*.

Otra opción que permite el filtro es la de obtener una muestra del conjunto de datos, pero manteniendo el peso que tenían las clases inicialmente, para esto se puede utilizar el parámetro *adjustWeights*, con el cual a las tuplas de un cada uno de las clases se les asignarán un peso determinado como se puede ver en la *figura 1.22*. Estos pesos podrían llegar a ser de utilidad en algunos clasificadores.

```
249,0.755298,0.475932,0.965569,0.635935,white,Ghoul,{1.102564}
287,0.479818,0.569762,0.721362,0.491938,blood,Ghoul,{1.102564}
292,0.476892,0.478592,0.40363,0.544587,blue,Ghoul,{1.102564}
240,0.49961,0.436475,0.591752,0.659622,clear,Ghoul,{1.102564}
139,0.416152,0.418087,0.802077,0.454922,clear,Ghoul,{1.102564}
17,0.585559,0.585939,1,0.708692,black,Ghoul,{1.102564}
264,0.51443,0.308411,0.533296,0.450843,white,Goblin,{1.068376}
348,0.355313,0.344684,0.555728,0.638232,blood,Goblin,{1.068376}
136,0.324769,0.27711,0.800646,0.359682,clear,Goblin,{1.068376}
269,0.403727,0.364366,0.609004,0.487301,white,Goblin,{1.068376}
```

Figura 1.22: Filtro SpreadSubSample utilizando el parámetro *adjustWeights*.

### 1.6.4. filters/supervised/instance/Resample

El filtro consiste en obtener un subconjunto de valores del *dataset* original mediante muestreo, ya sea con reemplazamiento o sin reemplazamiento, es decir, si se escoge una instancia, si el muestreo es con reemplazamiento, esta se podría volver a seleccionar, mientras que sin reemplazamiento no. El filtro, además, permite seleccionar (en %) cuántas tuplas se seleccionarán mediante muestreo del *dataset* original. Como diferencia con el filtro no supervisado, este permite desviar la distribución del nuevo *dataset* hacia una distribución más uniforme mediante la opción *biasToUniformClass*. Otra diferencia con respecto al filtro *unsupervised* es que este realiza el procedimiento de manera estratificada.

## 1.7. Actualización de la base de datos Criaturas Tenebrosas

- a) **Remove:** Se ha utilizado con el fin de eliminar el atributo id ya que no es relevante a la hora de realizar un proceso de clasificación.

| Current relation |  | Attributes: 7<br>Sum of weights: 371 |      |  |
|------------------|--|--------------------------------------|------|--|
| Attributes       |  |                                      |      |  |
|                  |  | All                                  | None |  |
| No.              |  | Name                                 |      |  |
| 1                | <input type="checkbox"/> id            |                                      |      |  |
| 2                | <input type="checkbox"/> bone_length   |                                      |      |  |
| 3                | <input type="checkbox"/> rotting_flesh |                                      |      |  |
| 4                | <input type="checkbox"/> hair_length   |                                      |      |  |
| 5                | <input type="checkbox"/> has_soul      |                                      |      |  |
| 6                | <input type="checkbox"/> color         |                                      |      |  |
| 7                | <input type="checkbox"/> type          |                                      |      |  |

Figura 1.23: Dataset antes de aplicar el filtro Remove.

| Current relation |  | Attributes: 6<br>Sum of weights: 371 |      |  |
|------------------|--|--------------------------------------|------|--|
| Attributes       |  |                                      |      |  |
|                  |  | All                                  | None |  |
| No.              |  | Name                                 |      |  |
| 1                | <input type="checkbox"/> bone_length   |                                      |      |  |
| 2                | <input type="checkbox"/> rotting_flesh |                                      |      |  |
| 3                | <input type="checkbox"/> hair_length   |                                      |      |  |
| 4                | <input type="checkbox"/> has_soul      |                                      |      |  |
| 5                | <input type="checkbox"/> color         |                                      |      |  |
| 6                | <input type="checkbox"/> type          |                                      |      |  |

Figura 1.24: Dataset después de aplicar el filtro NominalToBinary y Remove

- b) **NominalToBinary:** Se ha utilizado con el fin de que el atributo nominal *color* se transforme a seis atributos binarios diferentes con el fin de una mejor clasificación posterior.

| Current relation                           |                          | Attributes: 11      |        |         |
|--|--------------------------|---------------------|--------|---------|
| Relation: dataset371-weka.filters.unsup... |                          | Sum of weights: 371 |        |         |
| Attributes                                 |                          |                     |        |         |
|  | All                      | None                | Invert | Pattern |
| No.  |                          |                     | Name   |         |
| 1  | <input type="checkbox"/> | bone_length         |        |         |
| 2  | <input type="checkbox"/> | rotting_flesh       |        |         |
| 3  | <input type="checkbox"/> | hair_length         |        |         |
| 4  | <input type="checkbox"/> | has_soul            |        |         |
| 5  | <input type="checkbox"/> | color=clear         |        |         |
| 6  | <input type="checkbox"/> | color=green         |        |         |
| 7  | <input type="checkbox"/> | color=black         |        |         |
| 8  | <input type="checkbox"/> | color=white         |        |         |
| 9  | <input type="checkbox"/> | color=blue          |        |         |
| 10   | <input type="checkbox"/> | color=blood         |        |         |
| 11   | <input type="checkbox"/> | type                |        |         |

Figura 1.25: Dataset después de aplicar el filtro NominalToBinary y Remove.

Otros filtros que podrían resultar interesantes de aplicar serían el *RemoveUseless* y el *RemoveDuplicates* para eliminar aquellas instancias que no proporcionan ninguna información extra, sin embargo, en este *dataset* no realizan ningún cambio ya que no hay atributos irrelevantes ni instancias duplicadas, respectivamente.

## 2. PRÁCTICA 2. PARTE 1 – REGRESIÓN Y CLASIFICACIÓN WEKA

### 2.1. Algoritmo IBK para k=2 en el entorno EXPLORER

*Con su base de datos (criaturas tenebrosas), en el entorno EXPLORER utilice el algoritmo de clasificación IBK con un 3-fold crossvalidation. Use un valor de vecinos k=2 y deje por defecto el resto de parámetros (puede cambiarlos si se informa e investiga previamente sobre ellos).*

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

    Correctly Classified Instances      235           63.3423 %
    Incorrectly Classified Instances   136           36.6577 %
    Kappa statistic                   0.4465
    Mean absolute error              0.2474
    Root mean squared error          0.4178
    Relative absolute error          55.7041 %
    Root relative squared error     88.6698 %
    Total Number of Instances        371

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

    TP Rate   FP Rate   Precision   Recall   F-Measure   MCC   ROC Area   PRC Area   Class
    0,853     0,285     0,615      0,853    0,714       0,541  0,815      0,641     Ghoul
    0,464     0,240     0,496      0,464    0,479       0,228  0,668      0,471     Goblin
    0,573     0,031     0,893      0,573    0,698       0,626  0,901      0,769     Ghost
    Weighted Avg.   0,633     0,190     0,662      0,633    0,630       0,462  0,792      0,624

==== Confusion Matrix ====

    a   b   c   <-- classified as
110  19  0 |   a = Ghoul
  59  58  8 |   b = Goblin
  10  40  67 |   c = Ghost
```

Figura 2.1: Resultados del algoritmo IBK con k=2.

#### 2.1.1. Interprete la salida en cuanto a los valores resumen de las métricas que proporciona Weka.

Del apartado *Summary*, se puede destacar la tasa de clasificaciones correctas que en este caso es de un 63.3423% (235), mientras que, la tasa de clasificaciones incorrectas es de un 36.6577 % (136); por otro lado, el estadístico kappa toma un valor 0.4465, lo que quiere decir que, al estar por encima de 0, el resultado proporcionado es mejor que si se empleara un clasificador aleatorio. El valor máximo que toma este estadístico es 1.

El resto de métricas que aparecen están más relacionadas con un problema de regresión.

- 2.1.2. Tenga en cuenta si se clasifican bien todas las clases de su problema (TP Rate por clase). Comente este aspecto en función de la salida proporcionada por Weka (“Detailed Accuracy By Class”).

Los ratios de verdaderos positivos por clase para k=2 es:

- TP Rate *Ghoul* -> 0.853
- TP Rate *Goblin* -> 0.464
- TP Rate *Ghost* -> 0.573

Se puede afirmar que el clasificador es mejor que uno aleatorio, ya que la tasa de aciertos de todas las clases es superior a un 33.33%, sin embargo, la tasa de aciertos de la clase *Goblin* se encuentra por debajo de las otras dos.

- 2.1.3. Fíjese en la matriz de confusión y haga una interpretación de la misma.

```
==== Confusion Matrix ====
a   b   c   <- classified as
110 19  0 |  a = Ghoul
 59 58  8 |  b = Goblin
 10 40  67 |  c = Ghost
```

Figura 2.2: Matriz de confusión IBK con k=2.

Como ya se comentó previamente, se puede observar en la figura 2.2 como la clase *Ghoul* es la mejor clasificada con diferencia, sin embargo, ahora también se aprecia que las instancias mal clasificadas son únicamente clasificadas como *Goblin*.

En cuanto a las clasificaciones de las instancias de tipo *Goblin*, estas, al igual que ocurría con *Ghoul*, son clasificadas como *Ghoul* y como *Goblin* mayoritariamente, aunque, en este caso se aprecia que hay un mayor error pues hay un número de instancias igualmente clasificadas como *Ghoul* y *Goblin* por igual, lo cual indica que al clasificador le cuesta discernir entre estas dos clases.

Las instancias de tipo *Ghost* son clasificadas mayoritariamente como *Ghost*, no obstante, hay una gran cantidad de instancias etiquetadas como, principalmente, *Goblin* y, en menor medida, *Ghoul*.

- 2.1.4. Con el botón derecho del ratón sobre la lista de resultados del panel izquierdo puede acceder también a gráficas. Comente lo que considere necesario sobre “Visualize Classifier Errors” y “Visualize ThresholdCurve”.

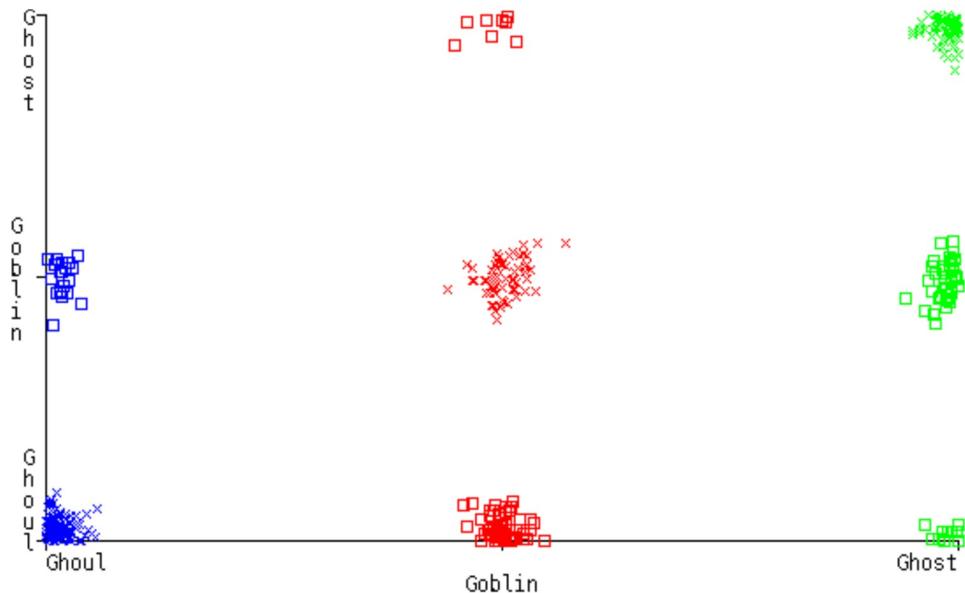


Figura 2.3: Visualize Classifier Errors sobre el modelo entrenado.

En *Visualize Classifier Errors* se puede apreciar una gráfica que se corresponde con la matriz de confusión que se ha estado comentando en el apartado anterior, con la salvedad que esta rotada 90º en sentido antihorario (ver Figura 2.3).

Con la opción *Visualize ThresholdCurve*, se pueden ver las curvas ROC para cada una de las clases.

*En este problema, el objetivo son tasas de acierto elevadas para cada uno de las clases, por ello, la curva ROC posiblemente no sea la mejor métrica a utilizar. La métrica más interesante, por lo tanto, es el TPrate por clase. Sin embargo, esto no quiere decir que la curva ROC no sea un valor representativo, ya que, con esta se denota, por ejemplo, como el clasificador es capaz de discernir mejor la clase Ghost del resto de clases.*

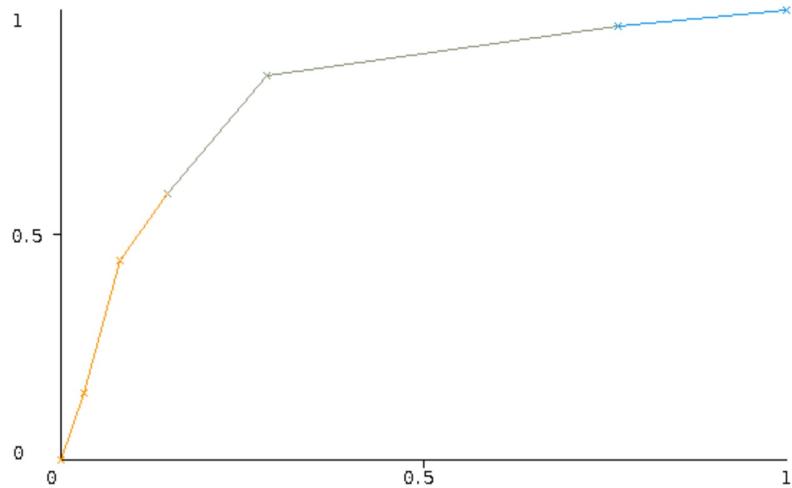


Figura 2.4: Curva ROC para la clase *Ghoul* IBK k=2.

Para la clase *Ghoul*, el valor que toma el área bajo la curva es 0,815. Se puede apreciar como al principio la relación entre el *TPrate* y el *FPrate*, es positiva, es decir, las instancias de tipo *Ghoul* que son clasificadas correctamente (en porcentaje) son mejor respecto a las muestras pertenecientes a otras clases y clasificadas como *Ghoul* (*FPrate*). El problema es que, a medida que va aumentando el umbral para el cual se considera que un patrón es positivo, el *FPrate* aumenta considerablemente, mientras que el *TPrate* crece de manera más lenta lo cual simboliza que la clasificación ya no es tan precisa (ver Figura 2.4).

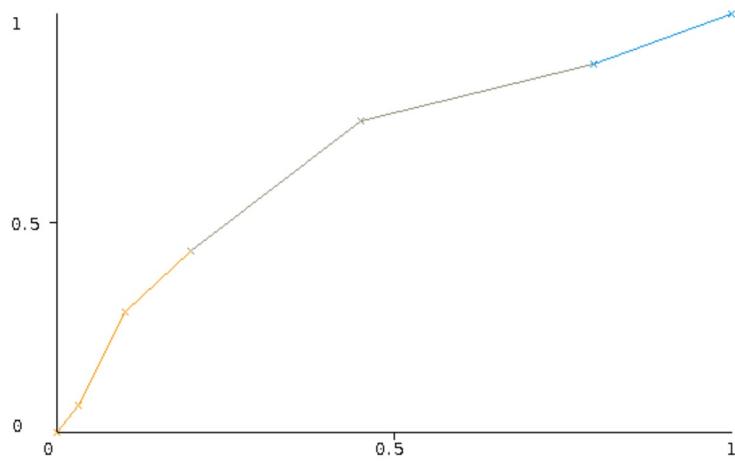


Figura 2.5: Curva ROC para la clase *Goblin* IBK k=2.

El área bajo la curva ROC (clase *Goblin*) es de 0,668. En este caso, la gráfica no tiene un pico tan pronunciado al principio, es decir, para valores bajos del umbral de discriminación, el *TPrate* toma valores bajos con respecto al *FPrate*. Esto disminuirá bastante el *AUC* y, por tanto, la calidad del clasificador indicada por esta métrica. Como se puede observar, el clasificador no es muy bueno para esta clase, debido a que se asemeja mucho a una línea recta, o lo que es lo mismo, a un *AUC* de 0,5, lo que indica que no discierne bien la clase *Goblin* de la del resto (ver Figura 2.5).

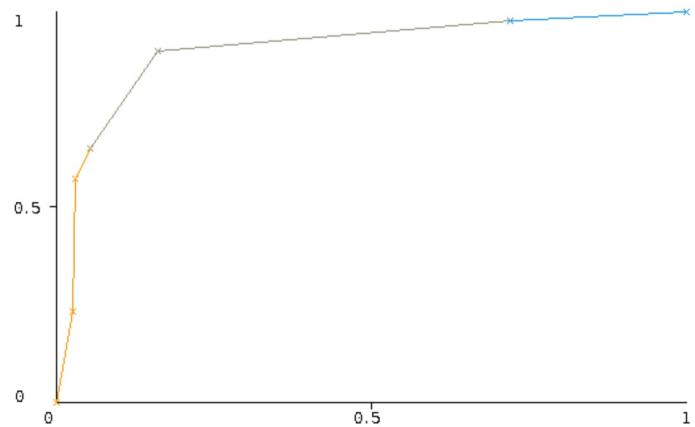


Figura 2.6 : Curva ROC para la clase Ghost IBK k=2.

El área bajo la curva *ROC* (*clase Ghost*) es 0,901. Esta es la gráfica con el mejor valor de *AUC* de las tres clases. Tal y como se puede apreciar para valores del umbral de discriminación bajo se tienen valores de *TPrate* por encima del *FPrate* que es lo que se persigue.

## 2.2. Algoritmo IBK para k=3 en el entorno EXPLORER

*Repetir los mismos ejercicios que en el apartado anterior y comparar los resultados.*

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

Correctly Classified Instances      231          62.2642 %
Incorrectly Classified Instances   140          37.7358 %
Kappa statistic                   0.4323
Mean absolute error               0.2607
Root mean squared error          0.4011
Relative absolute error           58.7045 %
Root relative squared error     85.119 %
Total Number of Instances        371

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

      TP Rate   FP Rate   Precision   Recall    F-Measure   MCC     ROC Area   PRC Area   Class
      0,667     0,277     0,562      0,667     0,610      0,377    0,826      0,682     Ghoul
      0,440     0,236     0,487      0,440     0,462      0,210    0,693      0,489     Goblin
      0,769     0,059     0,857      0,769     0,811      0,733    0,924      0,818     Ghost
Weighted Avg.      0,623     0,194     0,630      0,623     0,624      0,433    0,812      0,660

==== Confusion Matrix ====

      a   b   c   <-- classified as
86 42  1 |  a = Ghoul
56 55 14 |  b = Goblin
11 16 90 |  c = Ghost
```

Figura 2.7: Resultados del algoritmo IBK con k=3

- 2.2.1. Tenga en cuenta si se clasifican bien todas las clases de su problema (TP Rate por clase). Comente este aspecto en función de la salida proporcionada por Weka (“Detailed Accuracy By Class”).

Como se puede observar en la figura 2.7, el ratio de instancias correctamente clasificadas con el algoritmo *IBK* con k=3 es de un 62.26% (231) y el ratio de mal clasificadas es de un 37.73% (140). Con respecto al apartado anterior en el cual se usaba *IBK* con k=2, los resultados son bastante similares, un 63.34% (235) y un 36.65% respectivamente.

El estadístico *Kappa* tampoco varía significativamente, un valor de 0.4323 actualmente, frente a un valor, algo mejor, de 0.4465 en el apartado anterior.

- 2.2.2. Tenga en cuenta si se clasifican bien todas las clases de su problema (TP Rate por clase). Comente este aspecto en función de la salida proporcionada por Weka (“Detailed Accuracy By Class”).

| TP RATE %    |              |                         |
|--------------|--------------|-------------------------|
| Clase        | K=3 (Actual) | K=2 (Apartado anterior) |
| Ghoul        | 0.667        | 0.853                   |
| Goblin       | 0.440        | 0.464                   |
| Ghost        | 0.769        | 0.573                   |
| <i>Media</i> | 0.623        | 0.633                   |

En la tabla anterior, se aprecia una existente mejoría de las instancias de tipo *Ghost* correctamente clasificadas con k=3 en detrimento de las de tipo *Ghoul* que ven reducida su correcta clasificación con respecto a cuando se utilizó k=2. Por otro lado, las instancias de tipo *Goblin* mantienen su TP Rate bastante similar. Cabe decir que, la media de los bien clasificados se mantiene bastante semejante al caso anterior, dado que, el aumento de los bien clasificados de *Ghost* es proporcional a la reducción de correctamente clasificados de *Ghoul*.

- 2.2.3. Fíjese en la matriz de confusión y haga una interpretación de la misma.

```
==== Confusion Matrix ====
a b c    <- classified as
86 42 1 |  a = Ghoul
56 55 14 | b = Goblin
11 16 90 | c = Ghost
```

Figura 2.8: Matriz de confusión IBK con k=3.

Como ya se ha comentado en el subapartado anterior, la clase *Ghoul* (figura 2.8) se clasifica peor que para IBK k=2 (figura 2.2), aun así, la distribución de instancias se mantiene de forma similar, pues las instancias mal clasificadas son de tipo de *Goblin*. Por otro lado, las instancias de tipo *Goblin* se clasifican de manera casi idéntica. La clase que mejora es *Ghost* que clasifica más instancias

- 2.2.4. Con el botón derecho del ratón sobre la lista de resultados del panel izquierdo puede acceder también a gráficas. Comente lo que considere necesario sobre “Visualize Classifier Errors” y “Visualize ThresholdCurve”.

*Visualize Classifier Errors*, como ya se comentó en el apartado 2.1.4, se corresponde con la matriz de confusión, por lo que no se volverá a hablar de esta opción.

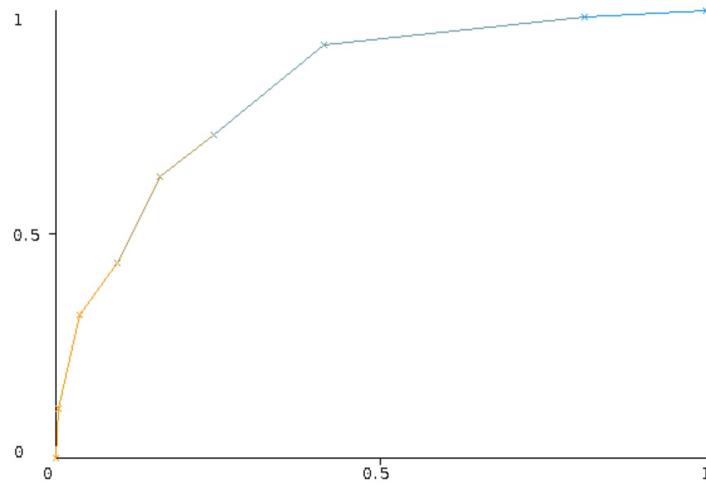


Figura 2.9: Curva ROC para la clase Ghoul IBK k=3.

El área bajo la curva ROC para *Ghoul* es de 0.82 (figura 2.9) frente a los 0.815 (figura 2.4), lo cual indica que tanto con *IBK* con  $k=2$ , como con  $k=3$ , la clase se clasifica igual, respecto a las muestras de *Ghoul* que se clasifican como *Ghoul* y respecto a las clases de otro tipo que se etiquetan como *Ghoul*.

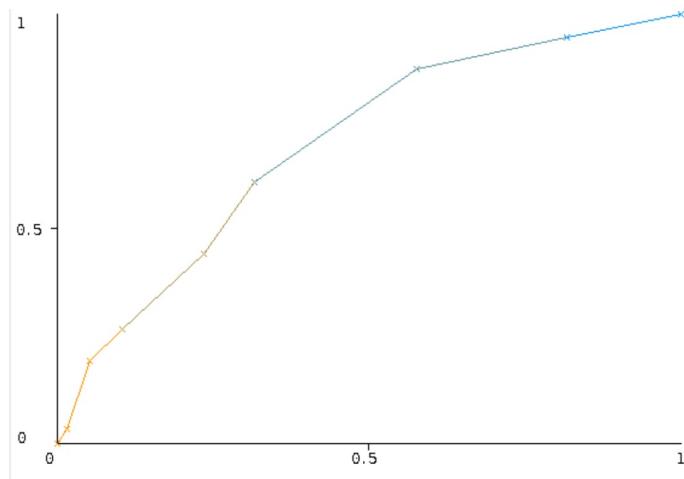


Figura 2.10: Curva ROC para la clase Goblin IBK k=3.

El área bajo la curva ROC para *Goblin* es de 0.69 (figura 2.10) frente a los 0.67 (figura 2.5), lo cual indica que tanto con *IBK* con  $k=2$ , como con  $k=3$ , de la misma manera que ocurría con la anterior comparación, se mantienen muy similares ambas curvas, aunque se denota una pequeña mejoría.

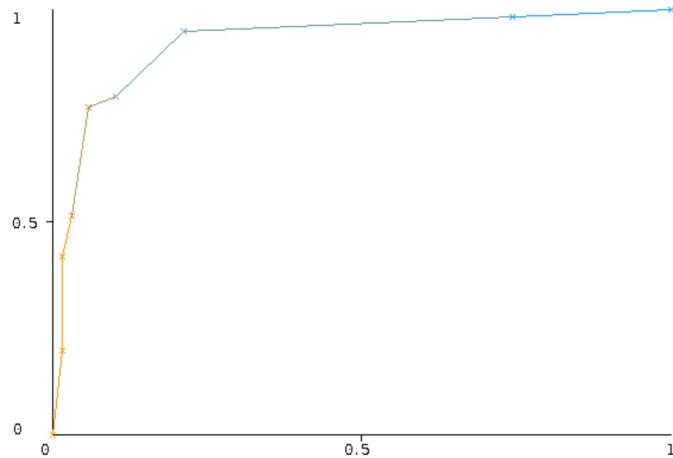


Figura 2.11: Curva ROC para la clase Ghost IBK  $k=3$ .

El área bajo la curva ROC para *Ghost* es de 0.92 (figura 2.11) frente a los 0.901 (figura 2.6), lo cual indica que tanto con *IBK* con  $k=2$ , como con  $k=3$ , de la misma manera que ocurría con la anterior comparación, se mantienen muy similares ambas curvas, aunque se denota una pequeña mejoría.

*Como conclusión de este subapartado, el área bajo la curva ROC se mantiene muy similar tras variar el número de vecinos para clasificar, aunque es cierto que con  $k=3$  se aprecia un mínimo aumento con respecto a  $k=2$ .*

### 2.3. Algoritmo IBK en el entorno EXPERIMENTER

Con su base de datos (criaturas tenebrosas), en el entorno *EXPERIMENTER* utilice el algoritmo de clasificación *IBK* con un *3-fold crossvalidation*. Use un valor de vecinos k igual a valor con el que haya obtenido los mejores resultados según el ejercicio anterior. Indique un número de repeticiones por *fold* igual a 1.

- 2.3.1. Calcule la media (en LibreOffice Calc es “PROMEDIO”) y la desviación típica de las medidas Accuracy, Kappa, RMSE, F-Measure. Configure las opciones de salida de manera que obtenga un fichero.csv que le permita calcular las medidas anteriores e interpretar los resultados.

| Percent_correct |
|-----------------|
| 65,3225806452   |
| 62,0967741935   |
| 62,6016260163   |
| 63,3403269517   |
| 1,7351415273    |

Figura 2.12: Accuracy

| Kappa_statistic |
|-----------------|
| 0,4761764417    |
| 0,4265472793    |
| 0,435892323     |
| 0,446205348     |
| 0,0263729464    |

Figura 2.13: Estadístico Kappa

| Root_mean_squared_error |
|-------------------------|
| 0,3874752059            |
| 0,4329520801            |
| 0,4315648071            |
| 0,4173306977            |
| 0,0258649169            |

Figura 2.14: RMSE

| F_measure    |
|--------------|
| 0,7115384615 |
| 0,7070707071 |
| 0,7238095238 |
| 0,7141395641 |
| 0,0086672543 |

Figura 2.15: F-Measure

Los valores numéricos que ocupan la cuarta y quitan posición (por filas), es decir, el penúltimo y el último si se comienza superiormente, se corresponden con la media y la desviación típica respectivamente.

- 2.3.2. ¿Se corresponde el (TP Rate por clase) en el fichero csv con los valores que observó en el ejercicio usando el EXPLORER? Comente y razoné la respuesta.

| True_positive_rate |
|--------------------|
| 0,8604651163       |
| 0,8139534884       |
| 0,8837209302       |
| 0,8527131783       |

Figura 2.16: Columna True positive rate del fichero resultado de aplicar KNN sobre el dataset Criaturas tenebrosas.

En este caso el EXPERIMENTER lo que muestra es, considerando como clase positiva la primera que aparece indicada en el dataset, el TP Rate que se obtiene cuando se entrena el modelo con cada uno de los 3 fold generados (Figura 2.16). En la figura también se ha añadido una fila que representa la media de los 3 valores. Por contra en EXPLORER lo que se muestra es la media de los TP rate que se obtiene en estos 3 fold, además de que esta media la va calculando para cada una de las clases (considera cada una de las clases como clase positiva).

- 2.3.3. ¿Qué diferencia habría si hiciera el mismo ejercicio indicando que el número de repeticiones por fold fuese igual a 3? ¿Qué es lo que cambiará en cada repetición?

| True_positive_rate |
|--------------------|
| 0.8604651162790697 |
| 0.813953488372093  |
| 0.8837209302325582 |
| 0.8604651162790697 |
| 0.8372093023255814 |
| 0.6976744186046512 |
| 0.7209302325581395 |
| 0.7906976744186046 |
| 0.8372093023255814 |

Figura 2.17: True positive\_rate para tres iteraciones con 3-fold.

En este caso se repetirá el mismo procedimiento que en el apartado anterior, pero 3 veces, es decir 3 filas correspondientes a cada repetición y cada una de estas 3 hace referencia a un fold.

Podría esperarse, por ejemplo, que la fila 2 y 5 tendrían que ser iguales ya que harían referencia al mismo fold, pero en una repetición distinta. Sin embargo, esto no es lo que ocurre, por lo tanto, la conclusión que se podría sacar es que el método que se emplea presenta una componente estocástica.

2.4. Usando el entorno EXPLORER ejecute el algoritmo Logistic con su base de datos, use un 3-fold crossvalidation como ya se hizo anteriormente.

2.4.1. Analice los modelos obtenidos, métricas, las variables que podrían ser más influyentes (valores beta), variables que no se usan, etc.

Una vez aplicado el algoritmo *Logistic*, una de las salidas de Weka son los coeficientes  $\beta_i$  de una regresión lineal para la función *logit* de la probabilidad (*Figura 2.18*).

$$g(x) = \ln \frac{\pi(x)}{1 - \pi(x)} = \beta_0 + \beta_1 x$$

*Figura 2.18: Función logit de la probabilidad*

| Coefficients... |          | Class    |        |
|-----------------|----------|----------|--------|
| Variable        |          | Ghoul    | Goblin |
| bone_length     | 16.2742  | 10.4018  |        |
| rotting_flesh   | -11.278  | -12.4957 |        |
| hair_length     | 21.5638  | 14.9056  |        |
| has_soul        | 18.3537  | 11.9192  |        |
| color=clear     | -0.8005  | -0.5326  |        |
| color=green     | 0.3077   | 0.3996   |        |
| color=black     | 0.4026   | 0.0741   |        |
| color=white     | 0.7838   | 0.4833   |        |
| color=blue      | 0.6337   | 1.3704   |        |
| color=blood     | -3.4715  | -3.5145  |        |
| Intercept       | -20.0206 | -9.2431  |        |

*Figura 2.19 :  $\beta_i$  para cada una de las variables.*

En el caso del algoritmo *Logistic*, se calculan los coeficientes de las regresiones lineales de dos de las clases (a través de máxima verosimilitud). A partir de estas regresiones lineales, aplicándoles una función *softmax* se puede obtener cual es la probabilidad de pertenencia de una determinada tupla a cada una de las clases.

$$p_1(\mathbf{x}, \hat{\theta}) = \frac{e^{f_1(\mathbf{x}, \hat{\theta})}}{1 + \sum_{k=1}^{K-1} e^{f_k(\mathbf{x}, \hat{\theta})}}$$

$$p_2(\mathbf{x}, \hat{\theta}) = \frac{e^{f_2(\mathbf{x}, \hat{\theta})}}{1 + \sum_{k=1}^{K-1} e^{f_k(\mathbf{x}, \hat{\theta})}}$$

*Figura 2.20: Valores de probabilidad de pertenencia a cada una mediante una función Softmax*

La probabilidad de pertenencia de la última clase será uno menos estas dos probabilidades. Pues el modelo, a partir de estas probabilidades, asignará cada una de las tuplas a la clase que tenga mayor probabilidad de pertenencia.

Para ver qué variables son más influyentes en el modelo, se utilizarán los *Odds ratios* (*Figura 2.21*). Estos *Odds ratios* se obtienen de  $e^{\beta_i}$ . La interpretación de estos es la siguiente:

Un valor mayor a 1 quiere decir que un aumento de esa variable independiente va a provocar un aumento en la probabilidad de la variable dependiente. Un valor menor que 1 significa que un aumento de la variable independiente provoca una disminución de la probabilidad de la variable dependiente. Por lo tanto, sabiendo esto se puede decir que las variables que más influye sobre la clase *Ghoul* es *hair\_length*, seguida de *has\_soul* y *bone\_length*. En este caso el modelo utiliza todas las variables, sin embargo, las menos influyentes serían *rotting\_flesh*, *color\_clear* y *color=blood*. En la clase *Goblin* las variables que influyen más y menos en el modelo son las mismas que en el caso anterior.

| Odds Ratios...       |                 | Class        |        |
|----------------------|-----------------|--------------|--------|
| Variable             |                 | Ghoul        | Goblin |
| <hr/>                |                 |              |        |
| <i>bone_length</i>   | 11689927.9473   | 32917.5773   |        |
| <i>rotting_flesh</i> | 0               | 0            |        |
| <i>hair_length</i>   | 2317694856.5821 | 2974672.6683 |        |
| <i>has_soul</i>      | 93517491.9764   | 150128.3578  |        |
| <i>color=clear</i>   | 0.4491          | 0.5871       |        |
| <i>color=green</i>   | 1.3603          | 1.4912       |        |
| <i>color=black</i>   | 1.4957          | 1.0769       |        |
| <i>color=white</i>   | 2.1899          | 1.6215       |        |
| <i>color=blue</i>    | 1.8845          | 3.9367       |        |
| <i>color=blood</i>   | 0.0311          | 0.0298       |        |

*Figura 2.21 : Odds ratios en el algoritmo Logistic.*

En cuanto a las métricas, se obtiene un *CCR* de 74.3935 % frente al 63.3423% que se obtuvo como máximo utilizando el clasificador *KNN*. El estadístico *kappa* (valores entre -1 y 1) toma un valor igual a 0.6156 lo que quiere decir que es mejor que un modelo que se base en puro azar. En cuanto a los *TPrate* por clase son 0,744, 0,624 y 0,872. Y el área bajo la curva *ROC* es igual a 0.903, 0.804 y 0.971 lo que quiere decir que se hace una buena distinción entre clases.

2.4.2. Asocie las fórmulas de las salidas por clase aportadas en las transparencias de esta práctica (transparencia titulada “Salidas por clase en Simplelogistic y Logistic de Weka”) con los modelos de probabilidad obtenidos en la salida de Weka.

Cada valor, respecto a cada clase, de la *figura 2.19*, se corresponde con un  $\beta_i$  del modelo de regresión para esa determinada clase y el atributo asociado, como se puede observar en la *figura 2.22*, donde  $X_i$ , es cada uno de los atributos.

$$f_1(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

Figura 2.22: Modelo de regresión de la clase 1

2.5. Haga lo mismo usando el algoritmo SimpleLogistic y compare resultados.

```

SimpleLogistic:

Class Ghoul :
-7.08 +
[bone_length] * 4.71 +
[rotting_flesh] * -1.78 +
[hair_length] * 5.93 +
[has_soul] * 4.98

Class Goblin :
2.4 +
[rotting_flesh] * -4.54 +
[hair_length] * 0.43 +
[color=white] * -0.09 +
[color=blue] * 0.29 +
[color=blood] * -0.49

Class Ghost :
7.21 +
[bone_length] * -5.82 +
[rotting_flesh] * 2.33 +
[hair_length] * -7.68 +
[has_soul] * -5.92

```

Figura 2.23: Modelos lineales resultantes para cada una de las clases.

En este caso la salida que proporciona *Weka*, se corresponde con las siguientes regresiones de la *diapositiva 37* de la práctica.

$$f_1(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i \quad f_2(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

$$f_3(\mathbf{x}, \hat{\theta}) = \hat{\beta}_0 + \sum_{i=1}^n \hat{\beta}_i x_i$$

Figura 2.24: Rectas de regresión

A partir de estas rectas, tal y como se ha indicado antes, se les aplicará una función *softmax* para obtener la probabilidad de pertenencia a cada una de las clases. Los valores  $\beta$  en este caso han sido calculados mediante una heurística.

En este caso las variables que aparecen en las rectas obtenidas son las que son relevantes para el modelo.

Con respecto a las métricas obtenidas se tiene un *CCR* de un 72.2372 % frente al 74.3935 % que daba el algoritmo de *Logistic*. El estadístico *KAPPA* es igual a 0.5833 que como en el caso anterior quiere decir que el modelo obtenido es mejor que uno basado en

azar. Por otro lado, los valores de *TPrate* por clase son 0.760, 0.560 y 0.855 que son bastante similares a los obtenidos en el apartado anterior. En cuanto al *AUC* se tiene valores 0.900, 0.799 y 0.970 lo que quiere decir que se discriminan bastante bien todas las clases igual que en apartado anterior.

### 3. Practica 2. Parte 2 – Clustering con Weka

3.1. Use el algoritmo K-means y seleccione la opción Use training set para la base de datos Iris. Para ello ignore el atributo de clase.

- Pruebe con 2 y 3 clusters analizando y discutiendo los clusters creados.

Para un número de *clusters* igual a 2 se obtiene un *SSE* total igual a 12.1437.

0 100 ( 67%)  
1 50 ( 33%)

Lo que aparece aquí reflejado son los dos *cluster* obtenidos y el número de patrones que pertenece a cada uno.

Para un número de *clusters* igual a 3 se obtiene un *SSE* total igual a 6.9981.

0 61 ( 41%)  
1 50 ( 33%)  
2 39 ( 26%)

El objetivo del *clustering* es minimizar el *SSE* total, es decir, la suma de las distancias al cuadrado de cada patrón de cada *cluster* con respecto a su centroide. Por lo tanto, basándose en esta métrica la ejecución del algoritmo *k-means* con un *k*=3 presenta un mejor rendimiento. Debido a que se conocía a priori el número de clases del *dataset*, era bastante lógico que el resultado obtenido para un *k*=3 iba a ser mejor.

- Para cada valor de *k* anterior, visualice los clusters resultantes al representar los atributos petallength y petalwidth y coméntelos.

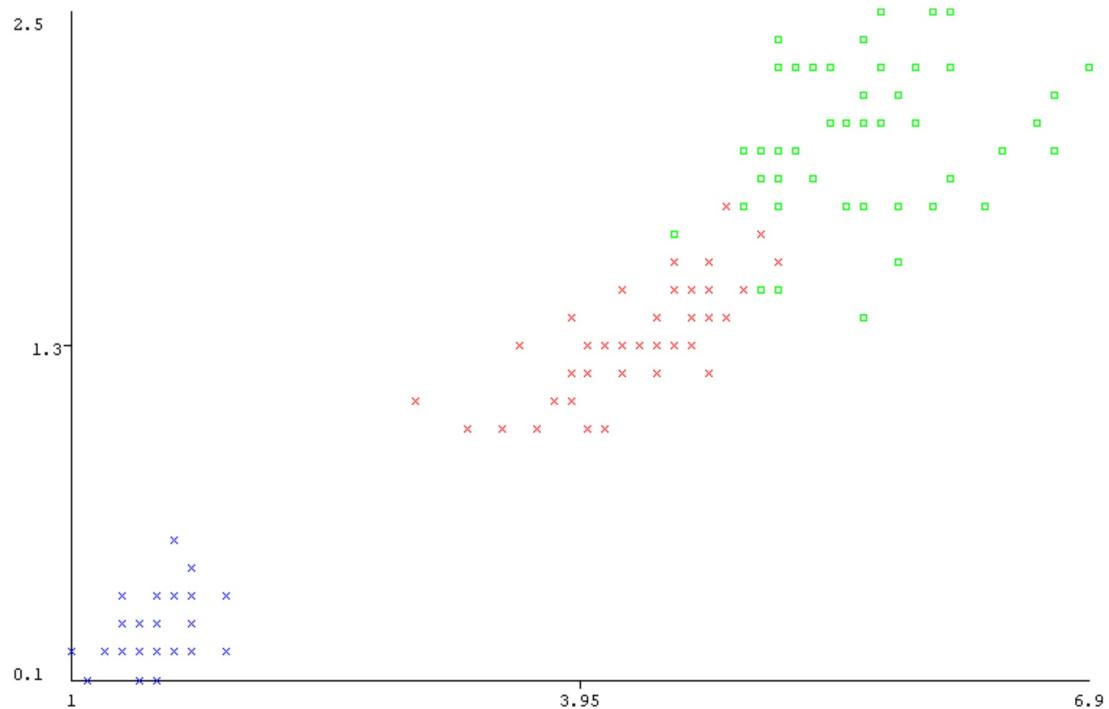


Figura 3.1: Representación de las clases del dataset Iris. X: petallength y Y: petalwidth.

Se ha incluido esta figura con la intención de comparar los *cluster* creados con respecto a la distribución original de las clases.

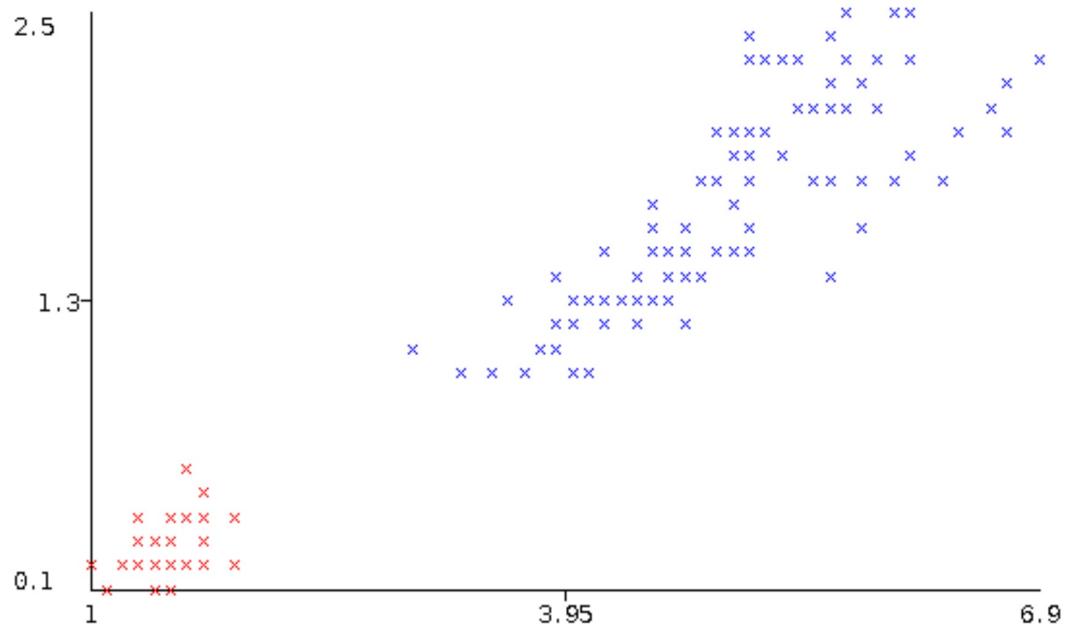


Figura 3.2: Clusters con el algoritmo K-Means con  $k=2$ . X: petallength y Y: petalwidth

Se puede apreciar en la *figura 3.1* que, de las tres clases que contiene la base de datos Iris, hay dos que están muy *correladas* y una muy *disjunta*, por ello, en la *figura 3.2*, al

ejecutar el algoritmo con dos *cluster*, el resultado es que, un *cluster* se crea entorno a las dos clases *correladas*, mientras que el otro se crea entorno a la clase disjunta.

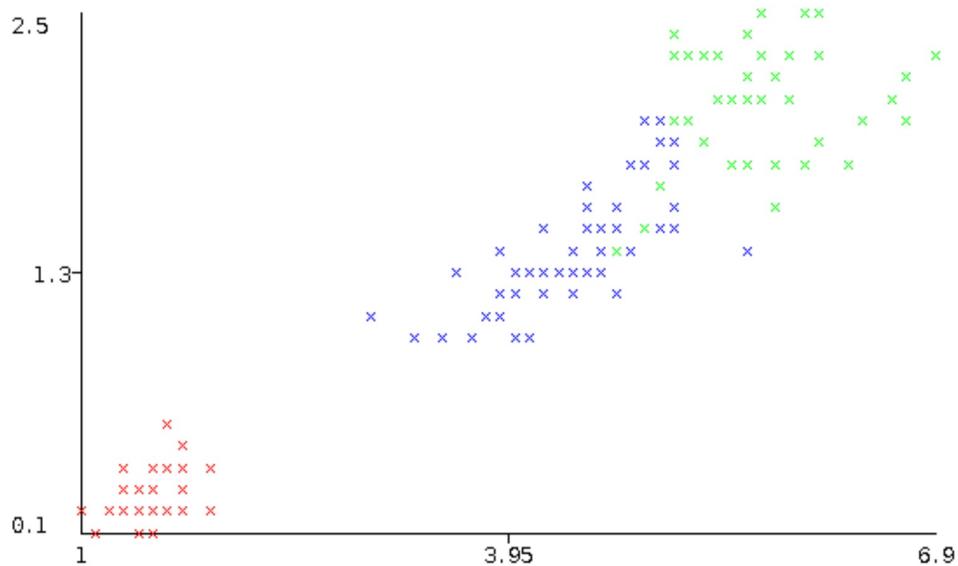


Figura 3.3: Clusters con el algoritmo K-Means con  $k=3$ . X: petallength y Y: petalwidth.

Tras ejecutar el algoritmo con tres *cluster*, el mismo número que clases existen, se puede apreciar (*Figura 3.3*) como la división presenta una forma similar, por no decir casi idéntica, que al visualizar la división de las clases en la *figura 3.1*. Esto era previsible dado que, se espera que el algoritmo de *clustering*, si conoce el número ideal de *clusters* (tres en este caso), obtenga una distribución similar a la que se tendría si existieran clases.

3.2. Cargue su base de datos de criaturas tenebrosas, seleccione la opción Use training set y analice qué ocurre al aplicar K-means, fijando  $k=\text{número-de-clases}$  de la base de datos. Discuta los clusters creados teniendo en cuenta:

- Considerando la etiqueta de clase.

Número de iteraciones: 10

*SSETotal*: 356.0997

|   |            |
|---|------------|
| 0 | 137 ( 37%) |
| 1 | 120 ( 32%) |
| 2 | 114 ( 31%) |

- Ignorando la etiqueta de clase.

Número de iteraciones: 4

*SSETotal:* 156.2052

0 137 ( 37%)  
 1 60 ( 16%)  
 2 174 ( 47%)

- Discusión

El valor del *SSETotal* cuando se tiene en cuenta la clase es mucho mayor que cuando se ignora este atributo, así como el número de iteraciones que tarda en converger es mayor. Esto se debe a que el atributo de la clase ha sido introducido al modelo, siendo este un atributo que añade información discordante, malogrando el modelo resultante. Como existe esta información que empeora el modelo, al algoritmo le cuesta más encontrar los centros, de ahí el número de iteraciones mayor.

- 3.3. ¿Qué ocurriría en K-means si fijásemos el número de clusters igual al número de patrones de una base de datos?, ¿De qué depende que K-means encuentre unos clusters u otros?

Lo que ocurriría es que existiría una relación 1 a 1 entre los centroides iniciales y cada uno de los patrones del *dataset*, es decir, cada uno de los *cluster* resultantes estarán formados por un único patrón.

Depende del número de *clusters* que se le indique al algoritmo y de la inicialización de los centroides, la cual va ligada a una semilla.

- 3.4. Utilizando su base de datos criaturas tenebrosas, el algoritmo K-means, y seleccionando la opción Classes to clusters evaluation, ¿con qué número de clusters obtiene mejores resultados?

- Para ello realice por cada valor de k 3 pruebas (cada una con una semilla diferente), y obtenga la media de la métrica SSE.

| Pruebas | K=2      | K=3      | K=4      | K=5      |
|---------|----------|----------|----------|----------|
| Seed=5  | 273.5173 | 157.6545 | 127.5313 | 68.1467  |
| Seed=10 | 210.4299 | 156.2052 | 91.3848  | 68.1467  |
| Seed=20 | 205.9072 | 129.2186 | 124.3381 | 117.6591 |
| Media   | 229.9515 | 147.6928 | 114.4181 | 84.6508  |

- Use la métrica de distancia que quiera, pero indique cuál ha usado.

Se ha usado la Distancia Euclídea.

- Represente en una gráfica el número de clusters frente al valor medio del SSE, y determine cuál es el valor de k más idóneo para su base de datos

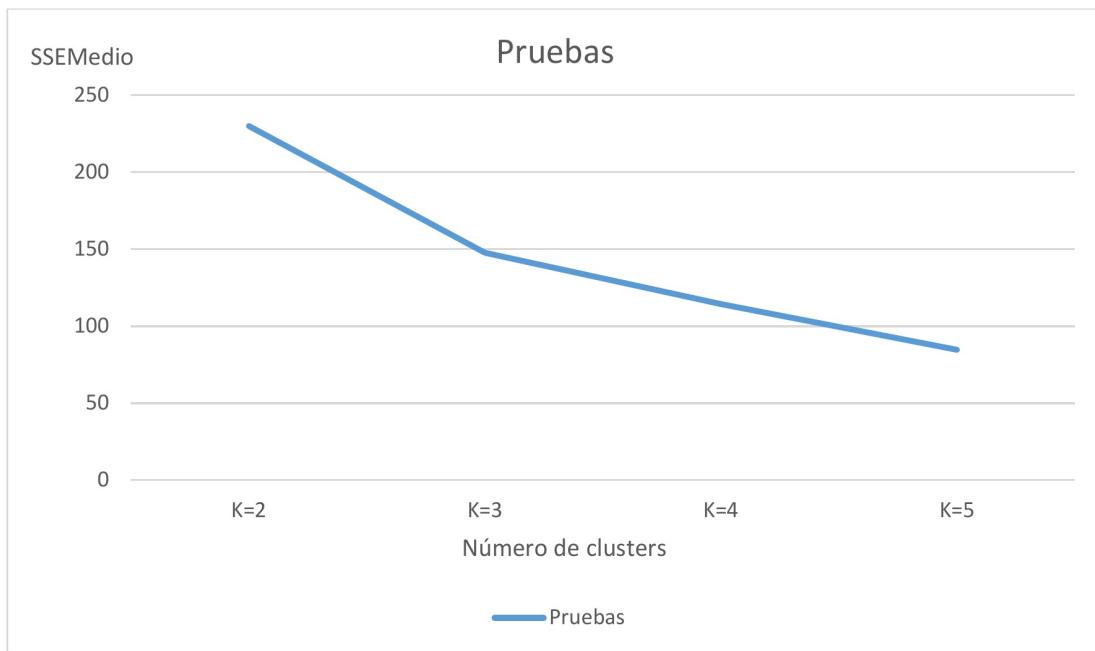


Figura 3.4: Relación entre el SSEMedio y el número de clusters para el algoritmo Kmeans.

El valor más idóneo es al aquél en el cual se encuentre el codo en la gráfica, en este caso, un valor de *clusters* = 3, el mismo que el número de clases original.

- 3.5. Cargue de nuevo la base de datos Iris, seleccione la opción Classes to clusters evaluation y ejecute el algoritmo HierarchicalClusterer con tipo de link Complete y métrica de distancia euclídea. Visualice las gráficas de los puntos agrupados. Comparando en el eje X instance\_number y el eje Y (atributos), vaya variando y mostrando cada uno de los atributos (sepallength, sepalwidth, petallength, petalwidth). ¿Cuáles de ellos producen los grupos mejor diferenciados y con fronteras claras?

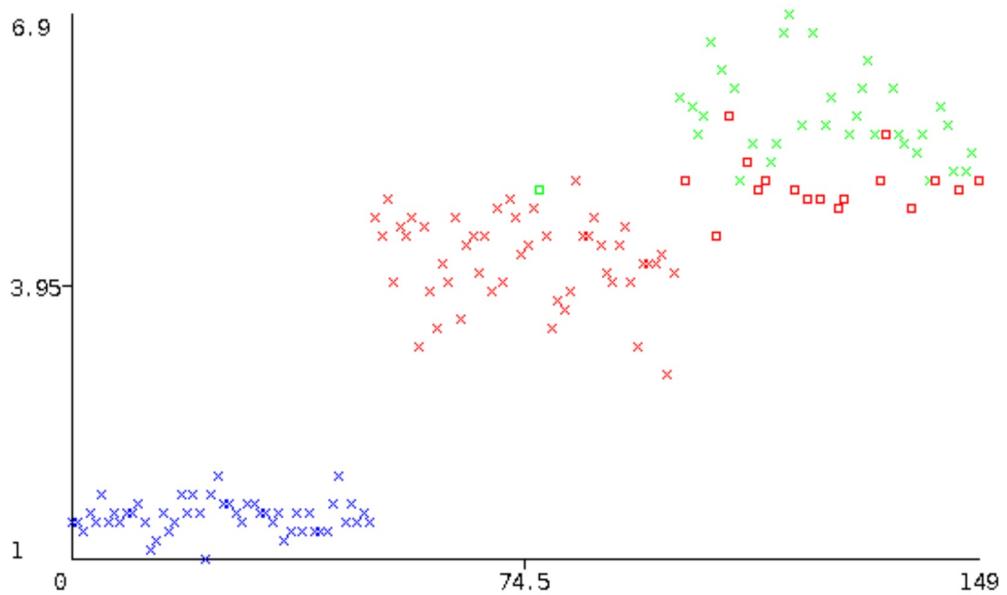


Figura 3.5: Eje X instance\_number, Eje Y petallength. HierarchicalClusterer

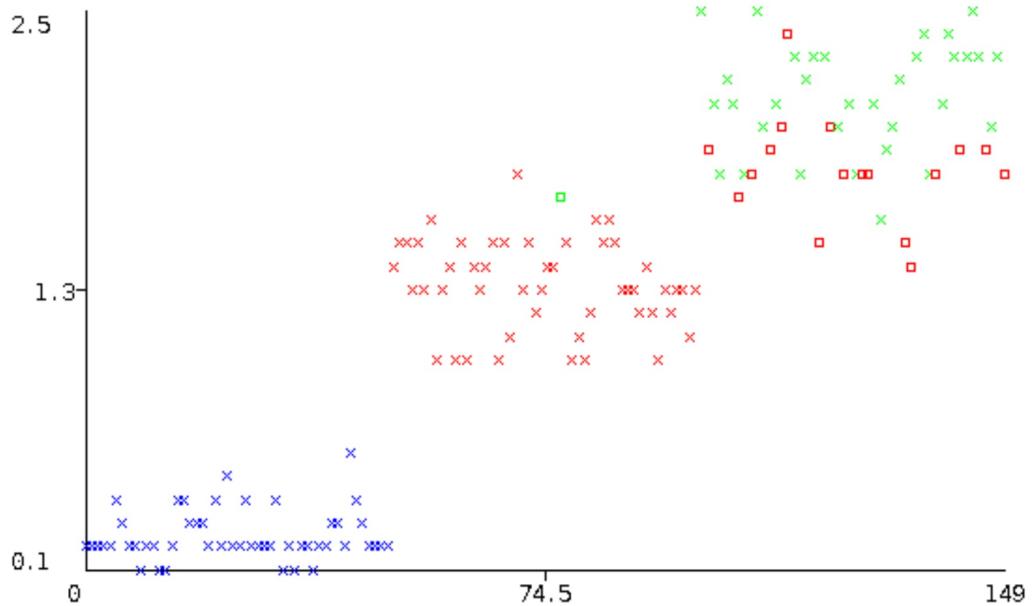


Figura 3.6: Eje X instance\_number, Eje Y petalwidth. HierarchicalClusterer

Tras observar todas las gráficas, el atributo que mejor separa los *clusters* es el *petallength* (figura 3.5), ya que se puede discernir una frontera clara que separa los dos *clusters* rojo y verde. Cabe mentar que, *petalwidth* también es un buen separador (figura 3.6), sin embargo, en su gráfica se aprecia como ambos *cluster* se encuentran algo mezclados, ya que, existen patrones que respecto al atributo *petalwidth* deberían pertenecer al *cluster* verde, no obstante, están agrupados en el *cluster* rojo.

3.6. Ejecute el algoritmo HierarchicalClusterer sobre su base de datos criaturas tenebrosas, seleccione la opción Classes to clusters evaluation, usando la distancia Euclidea e indicando como parámetro que se corte el dendograma en un número de clusters igual al número de clases de su problema.

- ¿Con qué linkType obtiene los mejores resultados en cuanto a instancias mal agrupadas? Pruebe con Single y Complete.

Basándose en el número de instancias mal agrupadas (*Incorrectly clustered instances*), donde con *linkType* igual a *Single* se tienen 239 y con *Complete* 238, se puede concluir que en el segundo caso los resultados obtenidos son mejores, aunque tampoco hay una gran diferencia. La diferencia principal va a estar en las asignaciones que se realizan tal y como se va a ver en el apartado siguiente.

- Analice los clusters creados con cada linkType y sus asignaciones.

#### linkType Single

```
Clustered Instances
0      340 ( 92%)
1      19 ( 5%)
2      12 ( 3%)

Class attribute: type
Classes to Clusters:

0  1  2 <- assigned to cluster
119 6  4 | Ghoul
116 7  2 | Goblin
105 6  6 | Ghost

Cluster 0 <- Ghoul
Cluster 1 <- Goblin
Cluster 2 <- Ghost

Incorrectly clustered instances :      239.0    64.4205 %
```

Figura 3.7: Base de datos Criaturas Tenebrosas, algoritmo HierarchicalClusterer con linkType Single.

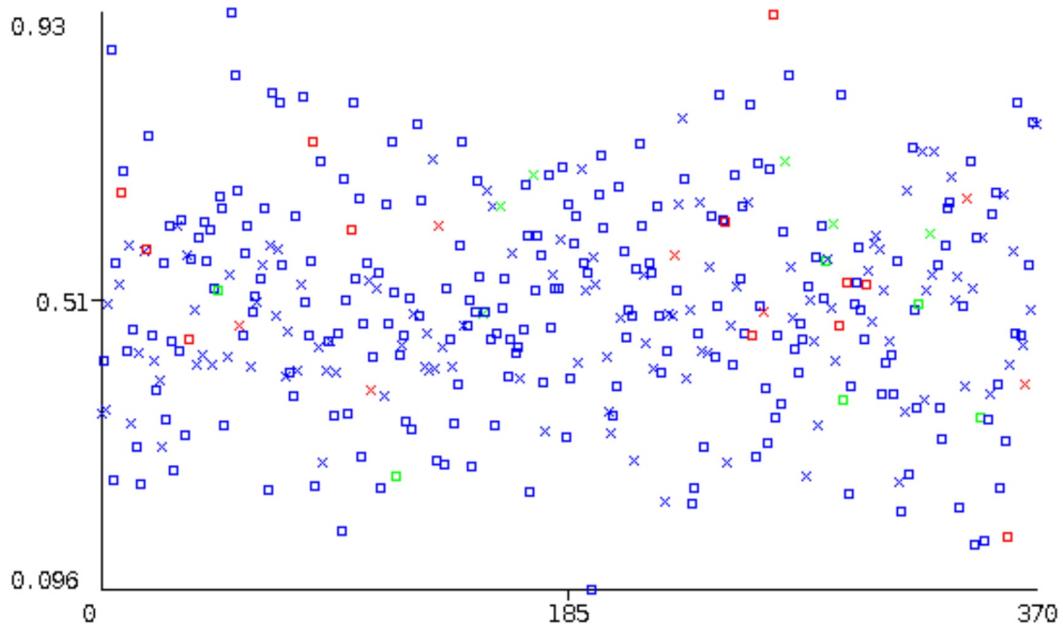


Figura 3.8: Eje X instance\_number, Eje Y rotting\_flesh.

Al observar el número de instancias mal agrupadas (*Figura 3.7*), se denota un alto grado de error (64.42 %), esto se debe a que prácticamente todas las instancias son asignadas al *cluster 0* (*clase Ghoul*), por lo tanto, al ser una base de datos balanceada y dos tercios de esta están mal agrupados, se tiene un error en torno al 66 %.

Al usarse esta distancia entre *clusters* (*Single*), la tendencia es que, si existen un número muy alto de instancias muy juntas en el espacio, estas se agruparán en único *cluster*, dejando el resto de *clusters* muy reducidos en cuanto a número de instancias contenidas. Por lo tanto, como ya se ha mencionado antes, se están agrupando casi todas las instancias como *Ghoul* debido a que los datos de esta base de datos están muy próximos.

En la *figura 3.8*, se observan como casi todas las instancias están agrupadas en un mismo *cluster* y que los datos están muy concentrados en torno al valor 0.5 de *rotting\_flesh*, es decir muy próximos entre sí.

### linkType Complete

```
Clustered Instances

0      132 ( 36%)
1      102 ( 27%)
2      137 ( 37%)

Class attribute: type
Classes to Clusters:

0 1 2 <- assigned to cluster
46 33 50 | Ghoul
48 34 43 | Goblin
38 35 44 | Ghost

Cluster 0 <- Goblin
Cluster 1 <- Ghost
Cluster 2 <- Ghoul

Incorrectly clustered instances :          238.0    64.1509 %
```

Figura 3.9: Base de datos Criaturas Tenebrosas, algoritmo HierarchicalClusterer con linkType Complete.

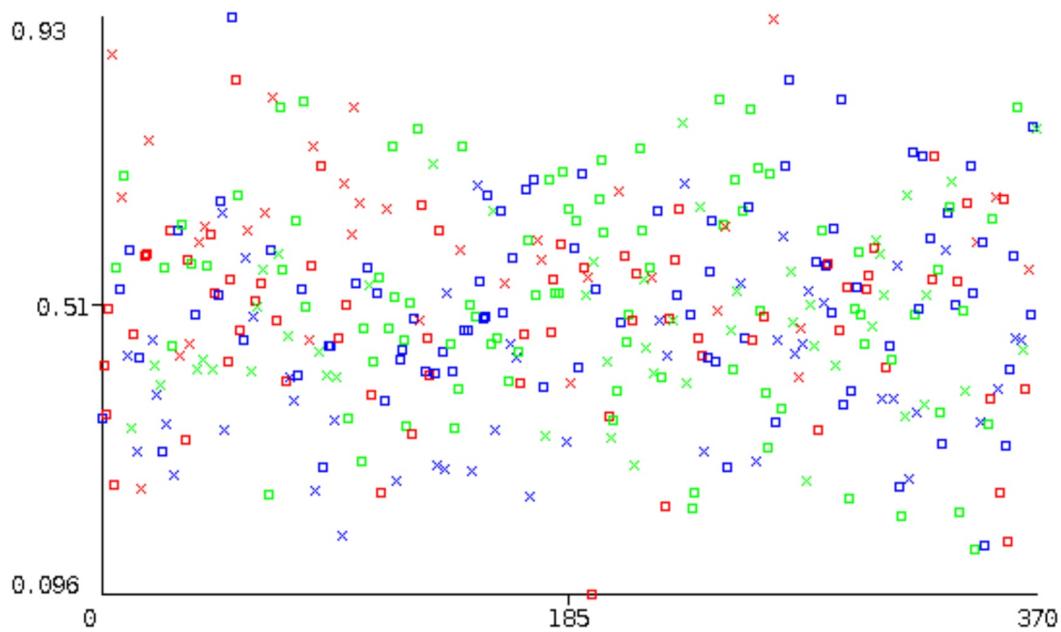


Figura 3.10: Eje X instance\_number, Eje Y rotting\_flesh.

El error, en este caso 64.15%, es muy semejante al del anterior, sin embargo, aquí el error no surge de un desbalance en los clústeres, donde prácticamente todas las instancias

formaban parte de un único *cluster*, sino que proviene de asignaciones desacertadas de los patrones, pero dando lugar a *clusters* más balanceados, como se puede denotar en *Clustered Instances* y en la matriz de confusión de la *figura 3.11*.

La diferencia de utilizar esta métrica de distancia (complete) con respecto a la anterior (single) es que, aquí, aunque todas las instancias se encuentren próximas entre sí, como lo que se tiene en cuenta para agrupar *clusters* es minimizar la distancia entre los dos patrones más distintos (más lejanos), se tenderá a crear *clusters* más equilibrados en cuanto a número de instancia, ya que cuantas más instancias posea un *cluster*, más probable es que exista una distancia mayor entre el *cluster* con más patrones y otro *cluster* cualquier, provocando que otros *cluster* se unan, creando así *clusters* balanceados.

## 4. Práctica 3. Árboles y redes neuronales

4.1. Cargue su base de datos y ejecute el algoritmo C4.5 usando un 75 % para entrenar y un 25 % para generalizar

4.1.1. 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.

La aplicación del algoritmo con los parámetros por defecto que proporciona weka, ha dado como resultado un número de nodos de 53, de los cuales, 27 son hojas.

Los atributos “color=green”, “color=blood” y “color=blue” no influyen mucho en el modelo, por ello, no se encuentran incluidos en el árbol como nodos separadores, mientras que el resto de atributos sí (“hair\_length”, “has\_soul”, “rotting\_flesh”, “bone\_length”, “color=black”, “color=white”, “color=clear”).

Cabe destacar que el nodo raíz del árbol es el atributo *hair\_length*, esto da a entender que, este atributo es el atributo que aporta más información.

Los valores resultantes del *TPRate* (*TPRate Ghoul: 0.767, TPRate Goblin: 0.676, TPRate Ghost: 0.690*) por clase tras aplicar el algoritmo son bastante similares, por lo que, se puede concluir que el árbol que se genera es capaz de diferenciar bastante bien entre las tres clases como también indica el área bajo la *curva ROC* (la media) que es 0.755. Con respecto a la métrica KAPPA se puede afirmar que el modelo obtenido es mejor que uno aleatorio ya que su valor se encuentra por encima de 0 (*Kappa statistic: 0.5619*).

Para finalizar, el *CCR* es de 70.9677%, por lo que, en general el modelo resultante es bueno.

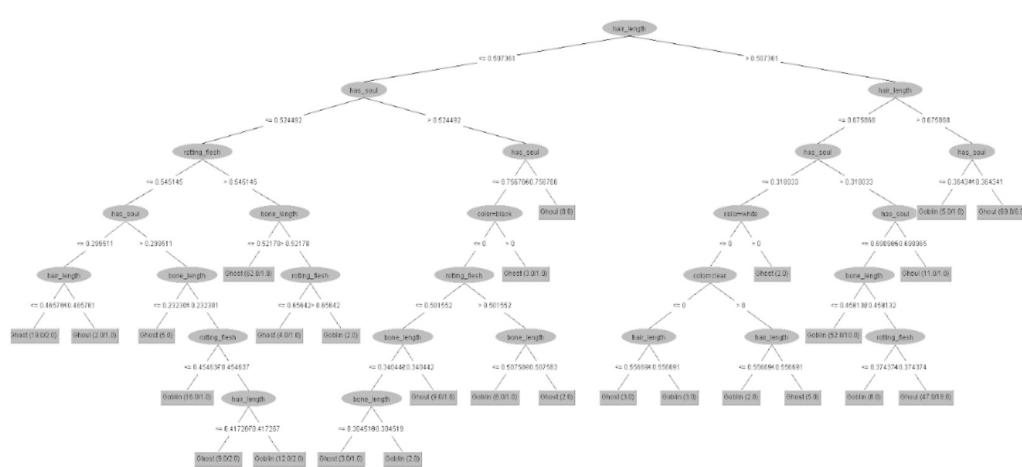


Figura 4.1: Árbol de decisión J.48 con parámetros por defecto de la base de datos Criaturas Tenebrosas.

Se ha incluido la figura 4.1 porque así se indicaba en el enunciado, sin embargo, esta no esclarece mucho las conclusiones debido a su dimensión.

- 4.1.2. Analice y muestre cómo cambia el árbol (nodo principal, tamaño, número de hojas, variables presentes y omitidas), al modificar los siguientes parámetros: *unpruned*, *confidenceFactor* y *minNumObj*. Comente también los resultados de las métricas obtenidas.

- *unpruned false*

| Parámetros<br>( <i>confidenceFactor</i> /<br><i>minNumObj</i> ) | Nodo<br>Principal | Tamaño | Número de<br>hojas | Variables<br>omitidas   |
|---|-------------------|--------|--------------------|---|
| <b>Conf=0.15 min=2</b>  | hair_length       | 45     | 23                 | color=green, color=blue,<br>color=blood, color=white                              |
| <b>Conf=0.30 min=2</b>  | hair_length       | 67     | 34                 | color=blue, color=blood   |
| <b>Conf=0.55 min=2</b>  | hair_length       | 81     | 41                 | color=blue, color=blood   |
| <b>Conf=0.15 min=10</b>   | hair_length       | 23     | 12                 | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |
| <b>Conf=0.30 min=10</b>   | hair_length       | 27     | 14                 | color=green, color=blue,<br>color=blood, color=white,<br>color=black              |
| <b>Conf=0.55 min=10</b>   | hair_length       | 27     | 14                 | color=green, color=blue,<br>color=blood, color=white,<br>color=black              |
| <b>Conf=0.15 min=30</b>   | hair_length       | 13     | 7                  | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |
| <b>Conf=0.30 min=30</b>   | hair_length       | 13     | 7                  | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |
| <b>Conf=0.55 min=30</b>   | hair_length       | 15     | 8                  | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |

Tabla 4.1: Características del árbol para los resultados de *unpruned false*, algoritmo J.48.

Las conclusiones que se pueden extraer de la *tabla 4.1* son:

1. Las variables relacionadas con el color de las criaturas, ya sea en mayor o menor medida, siempre quedan excluidas en el modelo, esto es debido a que no aportan gran información. El resto de variables siempre están incluidas.
2. Como era de esperar, al aumentar *confidenceFactor* (factor de poda, mayor valor -> menor poda) y *unpruned* tomar el valor *false* (se permite poda), el tamaño del árbol y, por consiguiente, el número de hojas también aumentan. Por ejemplo, cuando *conf=0.15* y *min=2* el tamaño del árbol es 45, mientras que cuando *conf=0.55* y *min=2* el tamaño es 81 como se puede ver en la respectiva tabla 4.1.
3. El tamaño del árbol también está relacionado con el parámetro *minNumObj*. A mayor valor, el tamaño del árbol disminuye, ya que se obliga a que el número de instancias por nodo hoja aumente, por lo tanto, abarcará antes todas las instancias

del problema. Por ejemplo, cuando  $conf=0.15$  y  $min=10$ , el tamaño del árbol es de 27, mientras que cuando  $conf=0.15$  y  $min=30$ , el tamaño es 13.

4. Destacar que  $confidenceFactor$  y  $minNumObj$  son contrapuestos en cuanto al tamaño del árbol, cuando el primero aumenta el tamaño del árbol aumenta, mientras que cuando el segundo aumenta el tamaño del árbol disminuye.
5. Se observa que el atributo  $hair\_length$  es siempre el nodo raíz del árbol, esto es debido a que este atributo es el que más cantidad de información aporta.

| Parámetros<br>(confidenceFactor/<br>minNumbObj) | CCR %   | TPRate<br>Ghoul | TPRate<br>Ghost | TPRate<br>Goblin | Kappa  |
|---|---------|-----------------|-----------------|------------------|--------|
| <b>Conf=0.15 min=2</b>                          | 73.1183 | 0.767           | 0.759           | 0.676            | 0.5954 |
| <b>Conf=0.30 min=2</b>                          | 67.6419 | 0.667           | 0.69            | 0.676            | 0.5122 |
| <b>Conf=0.55 min=2</b>                          | 66.667  | 0.667           | 0.69            | 0.647            | 0.4963 |
| <b>Conf=0.15 min=10</b>                         | 68.8172 | 0.767           | 0.724           | 0.588            | 0.5319 |
| <b>Conf=0.30 min=10</b>                         | 68.8172 | 0.767           | 0.724           | 0.588            | 0.5319 |
| <b>Conf=0.55 min=10</b>                         | 62.3656 | 0.767           | 0.724           | 0.412            | 0.4373 |
| <b>Conf=0.15 min=30</b>                         | 61.2903 | 0.900           | 0.690           | 0.294            | 0.4243 |
| <b>Conf=0.30 min=30</b>                         | 64.5161 | 0.767           | 0.690           | 0.500            | 0.4668 |
| <b>Conf=0.55 min=30</b>                         | 64.5161 | 0.767           | 0.690           | 0.500            | 0.4668 |

Tabla 4.2: Métricas para los resultados de unpruned false, algoritmo J.48. Asociado con la tabla 4.1.

Las conclusiones que se pueden extraer de la tabla 4.2 son:

1. Por norma general, cuanto mayor es la poda ( $confidenceFactor$ ) y menor el número de instancias por hoja ( $minNumObj$ ), los modelos obtenidos son mejores. Por ejemplo, para  $conf=0.15$  y  $min=10$ , el  $CCR$  es de 73.1183 % mientras que para  $conf=0.55$  y  $min=10$ , el  $CCR$  es de 62.3656 %.
2. El algoritmo, por lo general, balancea bastante bien las clasificaciones de las tres clases, sin embargo, un aumento del parámetro  $minNumObj$  suele provocar un beneficio en la clasificación de las clases *Ghoul* y *Ghost* en detrimento de la clase *Goblin*.
3. Con respecto al estadístico *Kappa*, el modelo obtenido siempre se encuentra por encima de un clasificador estocástico, además, normalmente se encuentra en torno a 0.5, bastante alejado del 0 (clasificador aleatorio).

- unpruned true

| Parámetros<br>(confidenceFactor/<br>minNumbObj) | Nodo<br>Principal | Tamaño | Número de<br>hojas | Variables<br>omitidas   |
|---|-------------------|--------|--------------------|---|
| <b>Conf=0.15 min=2</b>                          | hair_length       | 91     | 46                 | color=blood   |
| <b>Conf=0.30 min=2</b>                          | hair_length       | 91     | 46                 | color=blood   |
| <b>Conf=0.55 min=2</b>                          | hair_length       | 91     | 46                 | color=blood   |
| <b>Conf=0.15 min=10</b>                         | hair_length       | 27     | 14                 | color=green, color=blue,<br>color=blood, color=white,<br>color=black              |
| <b>Conf=0.30 min=10</b>                         | hair_length       | 27     | 14                 | color=green, color=blue,<br>color=blood, color=white,<br>color=black              |
| <b>Conf=0.55 min=10</b>                         | hair_length       | 27     | 14                 | color=green, color=blue,<br>color=blood, color=white,<br>color=black              |
| <b>Conf=0.15 min=30</b>                         | hair_length       | 15     | 8                  | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |
| <b>Conf=0.30 min=30</b>                         | hair_length       | 15     | 8                  | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |
| <b>Conf=0.55 min=30</b>                         | hair_length       | 15     | 8                  | color=green, color=blue,<br>color=clear, color=blood,<br>color=white, color=black |

Tabla 4.3: Características del árbol para los resultados de unpruned true, algoritmo J.48.

Las conclusiones que se pueden extraer de la *tabla 4.3* son similares a las de la *tabla 4.1*, salvo que ahora no existirá poda, por lo que el único parámetro influyente es *minNumObj*, solo se tendrá en cuenta una fila de cada tres, debido a las variaciones de conf.

La conclusión 1 de la *tabla 4.1* se cumple, destacando que cuando *min=2* (tamaño del árbol máximo dentro de la experimentación), se incluirán en el modelo todos los atributos salvo *color=blood* debido al tamaño del árbol.

Las conclusiones 3 y 5 de la *tabla 4.1* se cumplen de igual manera.

Por otro lado, las conclusiones 2 y 4, en las cuales influye el parámetro *confidenceFactor*, no son relevantes en este caso.

| Parámetros<br>(confidenceFactor/<br>minNumbObj) | CCR %   | TPRate<br>Ghoul | TPRate<br>Ghost | TPRate<br>Goblin | Kappa  |
|---|---------|-----------------|-----------------|------------------|--------|
| <b>Conf=0.15 min=2</b>                          | 65.5914 | 0.667           | 0.690           | 0.618            | 0.4809 |
| <b>Conf=0.30 min=2</b>                          | 65.5914 | 0.667           | 0.690           | 0.618            | 0.4809 |
| <b>Conf=0.55 min=2</b>                          | 65.5914 | 0.667           | 0.690           | 0.618            | 0.4809 |
| <b>Conf=0.15 min=10</b>                         | 62.3656 | 0.767           | 0.724           | 0.412            | 0.4373 |
| <b>Conf=0.30 min=10</b>                         | 62.3656 | 0.767           | 0.724           | 0.412            | 0.4373 |
| <b>Conf=0.55 min=10</b>                         | 62.3656 | 0.767           | 0.724           | 0.412            | 0.4373 |
| <b>Conf=0.15 min=30</b>                         | 64.5161 | 0.767           | 0.690           | 0.500            | 0.4668 |
| <b>Conf=0.30 min=30</b>                         | 64.5161 | 0.767           | 0.690           | 0.500            | 0.4668 |
| <b>Conf=0.55 min=30</b>                         | 64.5161 | 0.767           | 0.690           | 0.500            | 0.4668 |

Tabla 4.4: Métricas para los resultados de unpruned true, algoritmo J.48. Asociado con la tabla 4.3.

Como ocurría con la tabla inmediatamente superior, las conclusiones son similares a las extraídas previamente, en este caso en la *tabla 4.2*.

La conclusión 2 de la *tabla 4.2* es exactamente la misma que la que se puede extraer aquí; en cuanto a la 3, es bastante similar, exceptuando el hecho de que el estadístico kappa es algo inferior al no realizarse poda.

La conclusión 1 de la *tabla 4.2*, suprimiendo la influencia del parámetro *confidenceFactor*, es también la misma, es decir, cuanto más aumenta el *minObj* menor es el *CCR*.

Como conclusión general de ambos apartados, los resultados con poda son mejores que los se consiguen sin esta, como ya se comentó en la conclusión 1 de la *tabla 4.2*.

## 4.2. Utilizando su base de datos con un 75/25 % y el algoritmo MultilayerPerceptron.

### 4.2.1. ¿Cuál sería el valor por defecto para el atributo *hiddenLayers* en su base de datos?

Por defecto el número de capas oculta es  $(\text{nº atributos} + \text{nºclases})/2$ .

- 4.2.2. Con los valores por defecto, ¿qué observa al ir modificando solo el *learningRate*? Use tablas para mostrar los resultados. Realice una gráfica que muestre cómo cambia el valor de *Correctly Classified instances* al modificar el parámetro, de forma que localice cuándo se estanca el aprendizaje o incluso se empieza a sobreentrenar.

| <b>Learning/<br/>Seed</b> | <b>0.1</b>   | <b>0.2</b>   | <b>0.3</b>   | <b>0.4</b>   | <b>0.5</b>   | <b>0.6</b>   | <b>0.7</b>   | <b>0.8</b>   | <b>0.9</b>   | <b>1</b>     |
|---------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| <b>0</b>                  | 65.59        | 68.81        | 63.44        | 66.66        | 67.74        | 67.74        | 69.89        | 66.66        | 68.81        | 67.74        |
| <b>10</b>                 | 65.59        | 66.66        | 65.59        | 64.51        | 63.44        | 67.74        | 65.59        | 68.81        | 72.04        | 70.96        |
| <b>20</b>                 | 65.59        | 68.81        | 63.44        | 62.36        | 67.74        | 64.51        | 69.89        | 73.11        | 63.44        | 69.89        |
| <b>30</b>                 | 68.81        | 59.13        | 65.59        | 70.96        | 65.59        | 70.96        | 68.81        | 69.89        | 65.59        | 73.11        |
| <b>PROMEDIO</b>           | <b>66.39</b> | <b>65.86</b> | <b>64.51</b> | <b>66.12</b> | <b>66.12</b> | <b>67.74</b> | <b>68.54</b> | <b>69.62</b> | <b>67.47</b> | <b>70.43</b> |

Tabla 4.5: Pruebas del algoritmo MultilayerPerceptron variando el parámetro *learningRate*

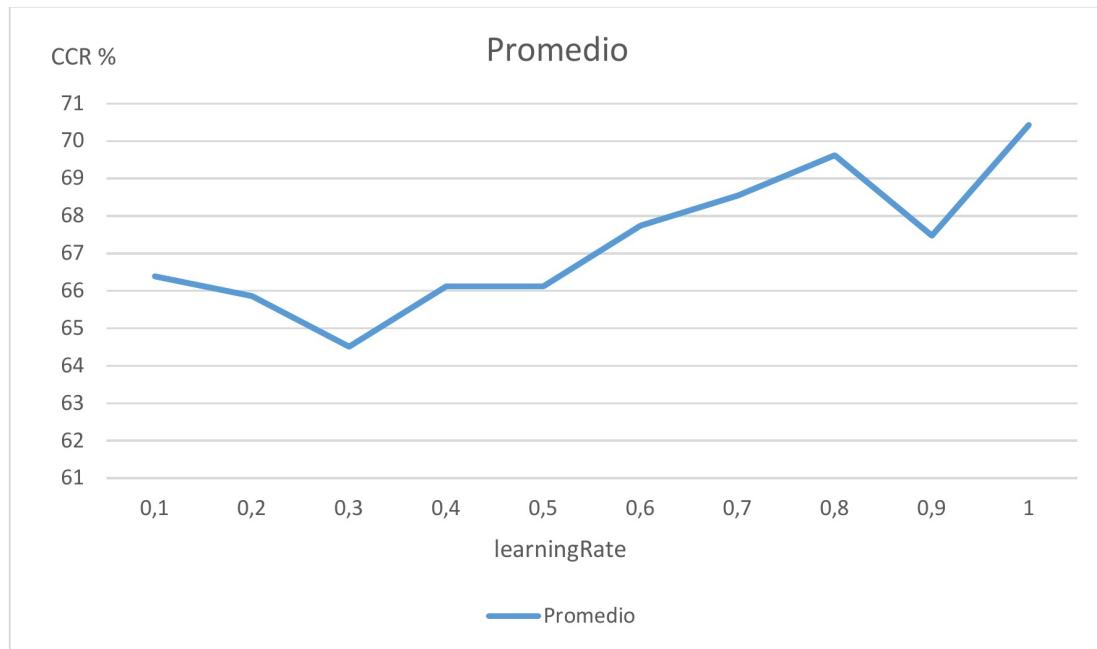


Figura 4.2: Gráfica obtenida a partir de las pruebas realizadas para *learningRate* Asociada a la tabla 4.5.

Con los datos obtenidos al modificar el parámetro *learningRate*, no se puede concluir en qué punto se comienza a sobreentrenar, ya que, por norma general el CCR aumenta conforme aumenta el *learningRate* como se puede observar en la figura 4.2, sin embargo, entre el valor de 0,8 y 0,9 de este parámetro se podría pensar que se produce un sobreentrenamiento debido a la brusca bajada del CCR, no obstante, esto no es así puesto que posteriormente con el valor 1 de *learningRate*, el CCR vuelve a aumentar, concluyendo que no se produce sobreentrenamiento. Por otro lado, en la gráfica no se observa ningún estancamiento del aprendizaje.

- 4.2.3. Con los valores por defecto, ¿qué observa al ir modificando solo el momentum? Use tablas para mostrar los resultados. Realice una gráfica que muestre cómo cambia el valor de Correctly Classified instances al modificar el parámetro, de forma que localice cuándo se estanca el aprendizaje o incluso se empieza a sobreentrenar.

| Momentum/<br>Seed | 0     | 0.1   | 0.2   | 0.3   | 0.4   | 0.5   | 0.6   | 0.7   | 0.8   | 0.9   |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0                 | 67.74 | 68.81 | 63.44 | 67.74 | 66.66 | 69.89 | 68.81 | 67.74 | 64.51 | 66.66 |
| 10                | 72.04 | 73.11 | 65.59 | 65.59 | 63.44 | 64.51 | 66.66 | 66.66 | 64.51 | 70.96 |
| 20                | 70.96 | 63.44 | 63.44 | 62.36 | 62.36 | 65.59 | 70.96 | 67.74 | 70.96 | 72.04 |
| 30                | 70.96 | 67.74 | 65.59 | 65.59 | 66.66 | 67.74 | 69.89 | 70.96 | 69.89 | 68.81 |
| PROMEDIO          | 70.43 | 68.27 | 64.51 | 65.32 | 64.78 | 66.93 | 69.08 | 68.27 | 67.47 | 69.62 |

Tabla 4.6: Pruebas del algoritmo MultilayerPerceptron variando el parámetro momentum.

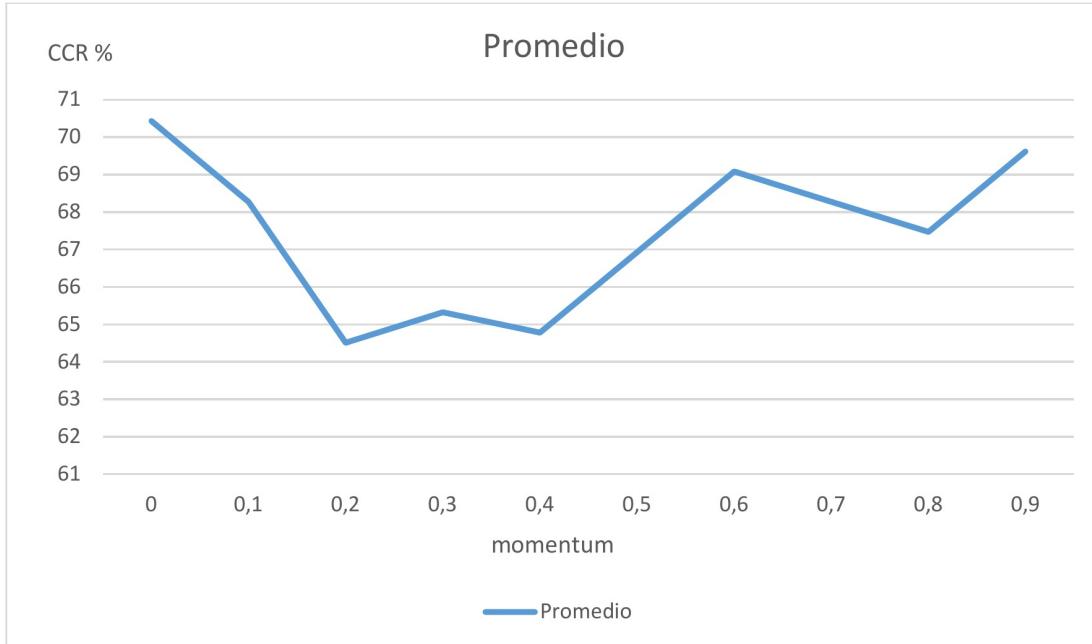


Figura 4.3: Gráfica obtenida a partir de las pruebas realizadas para momentum Asociada a la tabla 4.6.

No se pueden sacar conclusiones claras con respecto al sobreentrenamiento, ya que ni un aumento ni disminución del parámetro momentum lleva a un estancamiento o a un sobreentrenamiento en el aprendizaje del modelo, es decir, la gráfica no sigue una tendencia clara debido a que en ambos extremos se alcanzan los valores más altos de CCR. Destacar la zona en torno a los valores 0, 0,1 y 0,2 en la que se puede apreciar una bajada brusca en cuanto al CCR, sin embargo, a partir de 0,4 este vuelve a aumentar.

- 4.2.4. Con los valores por defecto, ¿qué observa al ir modificando solo el trainingTime? Realice una gráfica que muestre cómo cambia el valor de Correctly Classified instances al modificar el parámetro trainingTime, de forma que localice con cuántas épocas se estanca el aprendizaje o incluso se empieza a sobreentrenar.

| Time/<br>Seed | 250   | 500   | 750   | 1000  | 1250  | 1500  | 1750  | 2000  | 2250  | 2500  |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0             | 66.66 | 63.44 | 66.66 | 67.74 | 68.81 | 69.89 | 69.89 | 69.89 | 67.74 | 67.74 |
| 10            | 69.89 | 65.59 | 64.51 | 60.21 | 61.29 | 62.36 | 61.29 | 61.29 | 61.29 | 61.29 |
| 20            | 66.66 | 63.44 | 63.44 | 61.29 | 64.51 | 62.36 | 62.36 | 62.36 | 62.36 | 62.36 |
| 30            | 68.81 | 65.59 | 63.44 | 65.59 | 65.59 | 65.59 | 66.66 | 66.66 | 67.74 | 67.74 |
| PROMEDIO      | 68.01 | 64.51 | 64.51 | 63.70 | 65.05 | 65.05 | 65.05 | 65.05 | 64.78 | 64.78 |

Tabla 4.7: Pruebas del algoritmo MultilayerPerceptron variando el parámetro trainingTime.

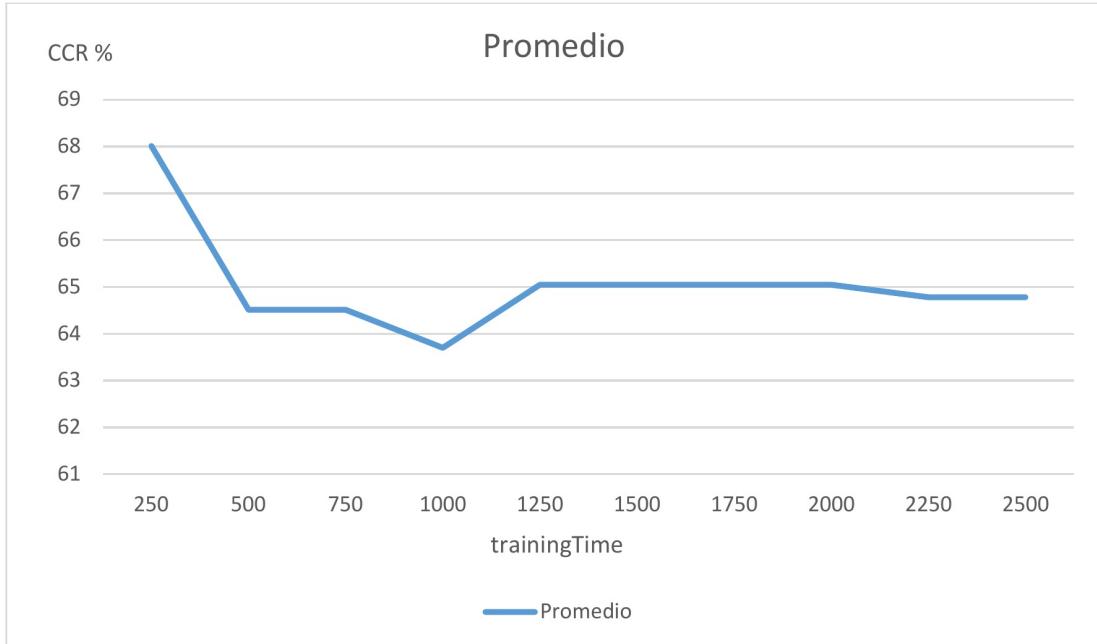


Figura 4.4: Gráfica obtenida a partir de las pruebas realizadas para trainingTime. Asociada a la tabla 4.7.

Aunque se podría llegar a pensar que se produce un estancamiento en el entrenamiento con los datos que aquí se muestran, esto no es del todo cierto. Al realizarse pruebas unitarias con diferentes valores elevados de *trainingTime* (50000, 70000, 101000, ...), se observó que el modelo seguía aprendiendo, ya que el valor del CCR obtenido seguía en aumento, no obstante, no tan considerablemente como se esperaba al aumentar en tan grandes cantidades el *trainingTime*.

## 5. Practica 4. Competición de Kaggle.

En esta sección, se realizarán diversos tratamientos sobre una base de datos proporcionada, por un lado, los datos de *train* (entrenamiento), por otro lado, los de *test* (generalización), los cuales aparecen sin el atributo de clase. Consecutivamente al tratamiento, se entrenará un modelo a partir de los datos de *train*. Con el fin de comprobar el modelo se clasificarán los patrones de *test* y el resultado de esta operación, será subida a la plataforma *Kaggle*, donde, como medida de calidad, se comprobará la métrica *FMeasure*.

El tratamiento de los se realizará tanto en *train* como en *test*, con la salvedad de que, todas las modificaciones realizadas en *test* serán a partir de los datos de *train*. Por ejemplo, para sustituir valores perdidos de un atributo en *test*, se utilizará la media de ese mismo atributo en *train*. Para no sobrecargar el documento con imágenes, se mostrarán, solo, los cambios producidos en *train*.

Cabe decir, que se han realizado múltiples pruebas con diferentes tratamientos sobre los datos y diferentes clasificadores, pero aquí solo se mostrará el tratamiento que permitió obtener la puntuación más alta en la plataforma *Kaggle*.

### 5.1. Tratamiento de datos perdidos

A lo largo de este apartado se comentará qué se ha hecho para paliar los datos perdidos. Inicialmente se eliminarán los atributos que tengan excesivos datos perdidos y, posteriormente, se reemplazarán los valores perdidos de aquellos atributos que solo posean un número aceptablemente bajo de valores perdidos, por la media.

| No. | Name       |
|-----|------------|
| 1   | air        |
| 2   | pres       |
| 3   | rhum       |
| 4   | uwnd       |
| 5   | vwnd       |
| 6   | WDIR       |
| 7   | WSPD       |
| 8   | GST        |
| 9   | DPD        |
| 10  | APD        |
| 11  | MWD        |
| 12  | PRES       |
| 13  | ATMP       |
| 14  | WTMP       |
| 15  | DEWP       |
| 16  | VIS        |
| 17  | TIDE       |
| 18  | Class_WVHT |

Figura 5.1: Estado inicial de la base de datos

Se muestra la *figura 5.1*, en la cual aparece el estado actual de la base de datos, con el fin de observar, más adelante, los cambios en la estructura.

### 5.1.1. Eliminación de atributos

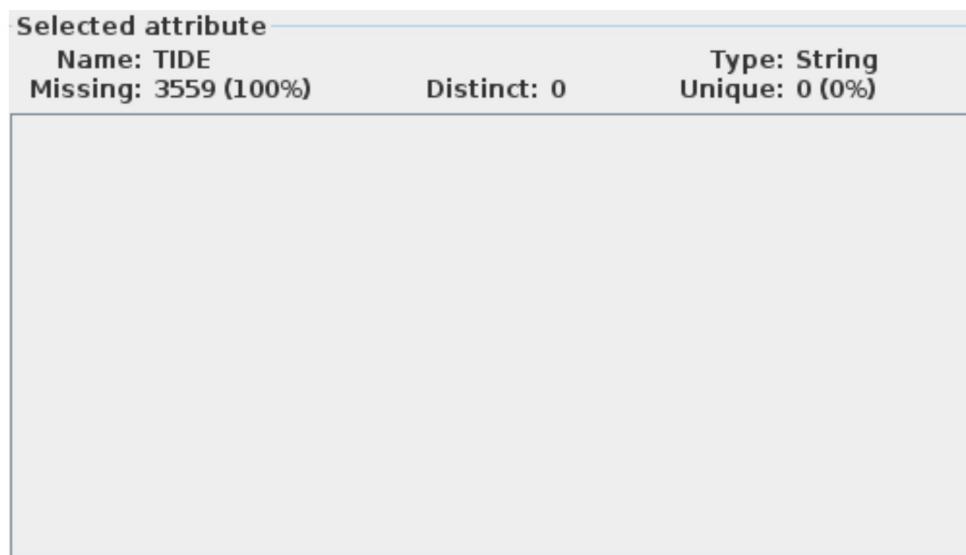


Figura 5.2: Valores perdidos del atributo TIDE.

El atributo *TIDE* tiene todos sus valores perdidos en el fichero de entrenamiento como se puede observar en la figura 5.2, se eliminó tanto del *train* como del *test*.

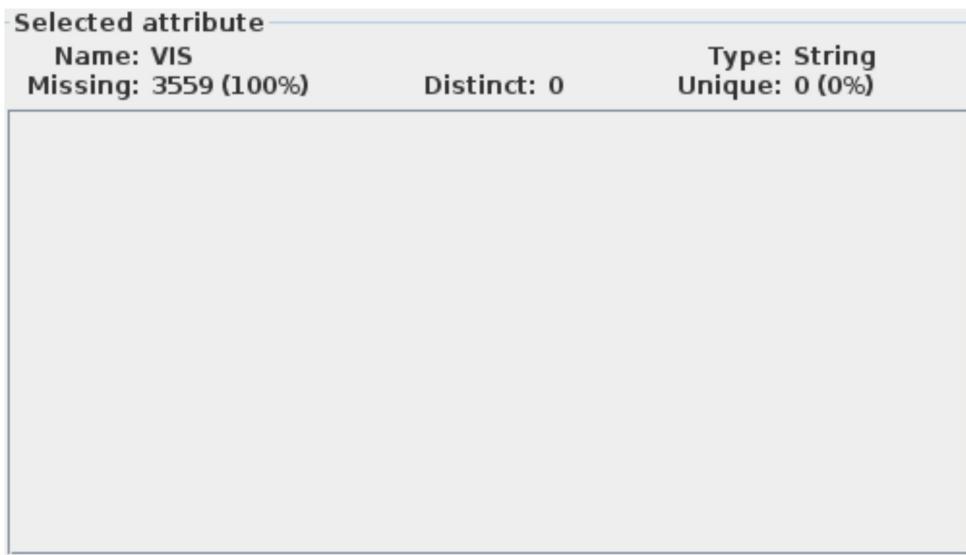


Figura 5.3:Valores perdidos del atributo VIS

El atributo *VIS* tiene todos sus valores perdidos en el fichero de entrenamiento como se puede observar en la figura 5.3, se eliminó tanto del *train* como del *test*.

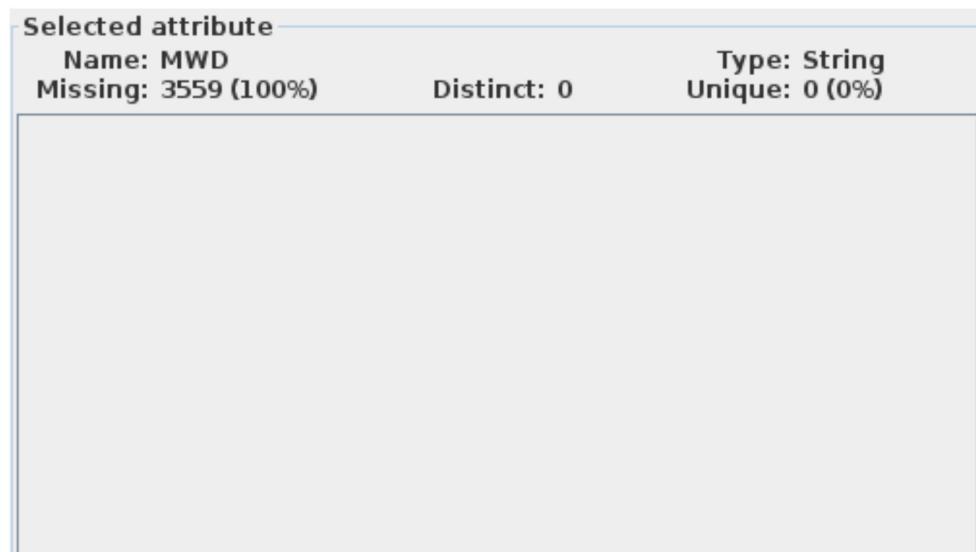


Figura 5.4: Valores perdidos del atributo MWD.

El atributo *MWD* tiene todos sus valores perdidos en el fichero de entrenamiento como se puede observar en la *figura 5.4*, se eliminó tanto del *train* como del *test*.

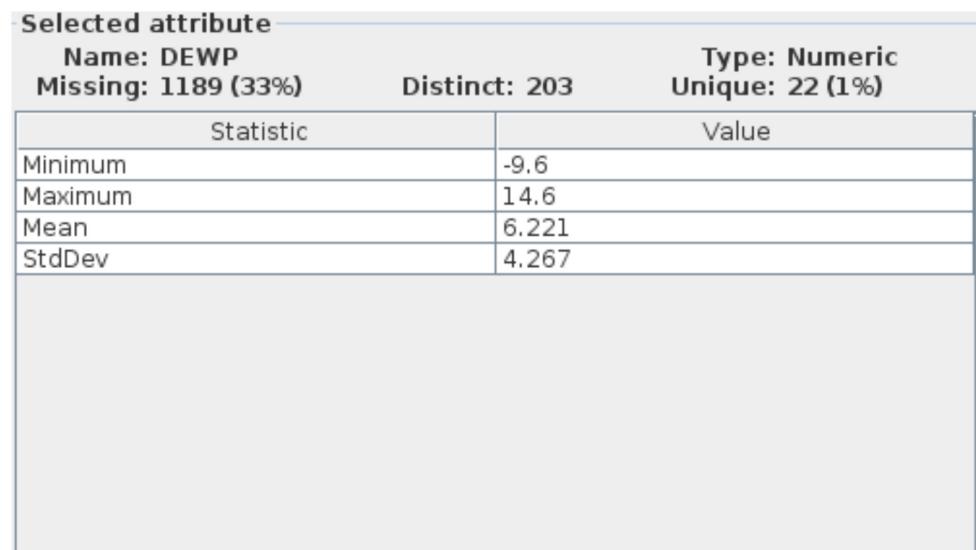


Figura 5.5: Valores perdidos del atributo DEWP.

Como se puede observar en la *figura 5.5*, el atributo *DEWP* tiene una cantidad considerable de valores perdidos (en torno al 33%). Tras múltiples pruebas, se observó que los resultados obtenidos al eliminar este atributo eran mejores que los que se obtenían considerándolo parte del modelo, de manera que se concluyó con su eliminación tanto en *train* como en *test*.

### 5.1.2. Recuperación de datos perdidos

Existen aún atributos que poseen valores perdidos, sin embargo, en menor medida. Para estos valores perdidos, el tratamiento dado es reemplazarlos por la media de su atributo correspondiente en *train*, tanto en el propio *train* como en el *test*. En esta sección, se mostrarán los cambios resultantes en *train* a través de imágenes, pero no en *test*, con el fin de no sobrecargar de figuras el documento.

| Name: APD  | Distinct: 551 | Type: Numeric   |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
|--|---------------|-----------------|---------------|-----------------|---------|-----------------|---|-------|------|-----------|--------|---------|-----|---------|-------|------|-------|--------|-------|----------------|--|--|
| Missing: 8 (0%)  |               | Unique: 90 (3%) |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| <table border="1"><thead><tr><th>Statistic</th><th>Value</th></tr></thead><tbody><tr><td>Minimum</td><td>3.1</td></tr><tr><td>Maximum</td><td>11.09</td></tr><tr><td>Mean</td><td>6.392</td></tr><tr><td>StdDev</td><td>1.189</td></tr></tbody></table>  |               |                 | Statistic     | Value           | Minimum | 3.1             | Maximum   | 11.09 | Mean | 6.392     | StdDev | 1.189   |     |         |       |      |       |        |       |                |  |  |
| Statistic  | Value         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Minimum  | 3.1           |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Maximum  | 11.09         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Mean   | 6.392         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| StdDev   | 1.189         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| <b>Antes</b>   |               |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| <table border="1"><tr><td>Name: APD</td><td>Distinct: 552</td><td>Type: Numeric</td></tr><tr><td>Missing: 0 (0%)</td><td></td><td>Unique: 90 (3%)</td></tr><tr><td colspan="3"><table border="1"><thead><tr><th>Statistic</th><th>Value</th></tr></thead><tbody><tr><td>Minimum</td><td>3.1</td></tr><tr><td>Maximum</td><td>11.09</td></tr><tr><td>Mean</td><td>6.392</td></tr><tr><td>StdDev</td><td>1.188</td></tr></tbody></table></td></tr><tr><td colspan="3"><b>Después</b></td></tr></table> | Name: APD     | Distinct: 552   | Type: Numeric | Missing: 0 (0%) |         | Unique: 90 (3%) | <table border="1"><thead><tr><th>Statistic</th><th>Value</th></tr></thead><tbody><tr><td>Minimum</td><td>3.1</td></tr><tr><td>Maximum</td><td>11.09</td></tr><tr><td>Mean</td><td>6.392</td></tr><tr><td>StdDev</td><td>1.188</td></tr></tbody></table> |       |      | Statistic | Value  | Minimum | 3.1 | Maximum | 11.09 | Mean | 6.392 | StdDev | 1.188 | <b>Después</b> |  |  |
| Name: APD  | Distinct: 552 | Type: Numeric   |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Missing: 0 (0%)  |               | Unique: 90 (3%) |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| <table border="1"><thead><tr><th>Statistic</th><th>Value</th></tr></thead><tbody><tr><td>Minimum</td><td>3.1</td></tr><tr><td>Maximum</td><td>11.09</td></tr><tr><td>Mean</td><td>6.392</td></tr><tr><td>StdDev</td><td>1.188</td></tr></tbody></table>  |               |                 | Statistic     | Value           | Minimum | 3.1             | Maximum   | 11.09 | Mean | 6.392     | StdDev | 1.188   |     |         |       |      |       |        |       |                |  |  |
| Statistic  | Value         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Minimum  | 3.1           |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Maximum  | 11.09         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| Mean   | 6.392         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| StdDev   | 1.188         |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |
| <b>Después</b>   |               |                 |               |                 |         |                 |   |       |      |           |        |         |     |         |       |      |       |        |       |                |  |  |

Figura 5.6: Reemplazamiento de valores perdidos en el atributo APD

El atributo *APD* tenía 8 valores perdidos (0%) en el *train* (Figura 5.6) y se reemplazaron por la media del *train* que era 6.392, tanto en *train* como en *test* (con la media de *train*).

| Name: DPD       | Distinct: 31 | Type: Numeric  |
|-----------------|--------------|----------------|
| Missing: 8 (0%) |              | Unique: 3 (0%) |
| <hr/>           |              |                |
| Statistic       | Value        |                |
| Minimum         | 2.94         |                |
| Maximum         | 17.39        |                |
| Mean            | 9.601        |                |
| StdDev          | 2.464        |                |

## Antes

| Name: DPD       | Distinct: 32 | Type: Numeric  |
|-----------------|--------------|----------------|
| Missing: 0 (0%) |              | Unique: 3 (0%) |
| <hr/>           |              |                |
| Statistic       | Value        |                |
| Minimum         | 2.94         |                |
| Maximum         | 17.39        |                |
| Mean            | 9.601        |                |
| StdDev          | 2.461        |                |

## Después

Figura 5.7: Reemplazamiento de valores perdidos en el atributo DPD

El atributo *DPD* tenía 8 valores perdidos (0%) en el *train* (Figura 5.7) y se reemplazaron por la media del *train* que era 9.601, tanto en *train* como en *test* (con la media de *train*).

| Name: PRES         | Distinct: 590 | Type: Numeric    |
|--------------------|---------------|------------------|
| Missing: 553 (16%) |               | Unique: 103 (3%) |
| <hr/>              |               |                  |
| Statistic          | Value         |                  |
| Minimum            | 958.8         |                  |
| Maximum            | 1038.3        |                  |
| Mean               | 1007.005      |                  |
| StdDev             | 13.499        |                  |

## Antes

| Name: PRES      | Distinct: 591 | Type: Numeric    |
|-----------------|---------------|------------------|
| Missing: 0 (0%) |               | Unique: 103 (3%) |
| <hr/>           |               |                  |
| Statistic       | Value         |                  |
| Minimum         | 958.8         |                  |
| Maximum         | 1038.3        |                  |
| Mean            | 1007.005      |                  |
| StdDev          | 12.406        |                  |

## Después

Figura 5.8: Reemplazamiento de valores perdidos en el atributo PRES

El atributo *PRES* tenía 553 valores perdidos (16%) en el *train* (Figura 5.8) y se reemplazaron por la media del *train* que era 1007, tanto en *train* como en *test* (con la media de *train*).

### 5.1.3. Establecer mismas unidades

En este apartado se unificarán las medidas para los mismos tipos de atributos, es decir, aquellos atributos que representen medidas similares y se encuentren en diferentes unidades de medida, se transformarán para que tengan una misma unidad de medida como puede ser la presión que está medida en Pascales y en Atmósferas. Asimismo, los cambios se producirán tanto en *train* como en *test*.

| Name: ATMP  | Distinct: 152 | Type: Numeric   |           |       |         |        |         |        |      |         |        |       |
|---|---------------|-----------------|-----------|-------|---------|--------|---------|--------|------|---------|--------|-------|
| Missing: 0 (0%)   |               | Unique: 16 (0%) |           |       |         |        |         |        |      |         |        |       |
| <table border="1"><thead><tr><th>Statistic</th><th>Value</th></tr></thead><tbody><tr><td>Minimum</td><td>-0.9</td></tr><tr><td>Maximum</td><td>15.8</td></tr><tr><td>Mean</td><td>9.238</td></tr><tr><td>StdDev</td><td>3.348</td></tr></tbody></table>       |               |                 | Statistic | Value | Minimum | -0.9   | Maximum | 15.8   | Mean | 9.238   | StdDev | 3.348 |
| Statistic   | Value         |                 |           |       |         |        |         |        |      |         |        |       |
| Minimum   | -0.9          |                 |           |       |         |        |         |        |      |         |        |       |
| Maximum   | 15.8          |                 |           |       |         |        |         |        |      |         |        |       |
| Mean  | 9.238         |                 |           |       |         |        |         |        |      |         |        |       |
| StdDev  | 3.348         |                 |           |       |         |        |         |        |      |         |        |       |
| <b>Antes</b>  |               |                 |           |       |         |        |         |        |      |         |        |       |
| Name: ATMP  | Distinct: 152 | Type: Numeric   |           |       |         |        |         |        |      |         |        |       |
| Missing: 0 (0%)   |               | Unique: 16 (0%) |           |       |         |        |         |        |      |         |        |       |
| <table border="1"><thead><tr><th>Statistic</th><th>Value</th></tr></thead><tbody><tr><td>Minimum</td><td>272.25</td></tr><tr><td>Maximum</td><td>288.95</td></tr><tr><td>Mean</td><td>282.388</td></tr><tr><td>StdDev</td><td>3.348</td></tr></tbody></table> |               |                 | Statistic | Value | Minimum | 272.25 | Maximum | 288.95 | Mean | 282.388 | StdDev | 3.348 |
| Statistic   | Value         |                 |           |       |         |        |         |        |      |         |        |       |
| Minimum   | 272.25        |                 |           |       |         |        |         |        |      |         |        |       |
| Maximum   | 288.95        |                 |           |       |         |        |         |        |      |         |        |       |
| Mean  | 282.388       |                 |           |       |         |        |         |        |      |         |        |       |
| StdDev  | 3.348         |                 |           |       |         |        |         |        |      |         |        |       |
| <b>Después</b>  |               |                 |           |       |         |        |         |        |      |         |        |       |

Figura 5.9: Transformación de unidades de ATMP.

El atributo *ATMP* es semejante al atributo *AIR*, ambos miden temperaturas. El atributo se mide en *Grados Celsius* mientras que *AIR* en *Kelvin*, por ello, el atributo *ATMP* se transforma a *Grados Kelvin* (Figura 5.9).

|            |               |                                   |
|------------|---------------|-----------------------------------|
| Name: PRES | Distinct: 591 | Type: Numeric<br>Unique: 103 (3%) |
| Statistic  | Value         |                                   |
| Minimum    | 958.8         |                                   |
| Maximum    | 1038.3        |                                   |
| Mean       | 1007.005      |                                   |
| StdDev     | 12.406        |                                   |

## Antes

|            |               |                                   |
|------------|---------------|-----------------------------------|
| Name: PRES | Distinct: 591 | Type: Numeric<br>Unique: 103 (3%) |
| Statistic  | Value         |                                   |
| Minimum    | 95880         |                                   |
| Maximum    | 103830        |                                   |
| Mean       | 100700.489    |                                   |
| StdDev     | 1240.612      |                                   |

## Después

Figura 5.10: Transformación de unidades de PRES.

El atributo *PRES* es semejante al atributo *pres*, ambos miden presiones. El atributo se mide en *Hectopascales* mientras que *pres* en *pascales*, por ello, el atributo *PRES* se transforma a *pascales* (Figura 5.10).

|            |               |                                 |
|------------|---------------|---------------------------------|
| Name: WTMP | Distinct: 102 | Type: Numeric<br>Unique: 3 (0%) |
| Statistic  |               | Value                           |
| Minimum    | 6             |                                 |
| Maximum    | 16.1          |                                 |
| Mean       | 10.146        |                                 |
| StdDev     | 3.097         |                                 |

## Antes

|            |               |                                 |
|------------|---------------|---------------------------------|
| Name: WTMP | Distinct: 102 | Type: Numeric<br>Unique: 3 (0%) |
| Statistic  |               | Value                           |
| Minimum    | 279.15        |                                 |
| Maximum    | 289.25        |                                 |
| Mean       | 283.296       |                                 |
| StdDev     | 3.097         |                                 |

## Después

Figura 5.11: Transformación de unidades de WTMP.

El atributo *WTMP* es semejante al atributo *AIR*, ambos miden temperaturas. El atributo se mide en *Grados Celsius* mientras que *AIR* en *Kelvin*, por ello, el atributo *WTMP* se transforma a *Grados Kelvin* (Figura 5.11).

#### 5.1.4. Eliminación de correlaciones

|  |  |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
|--|--|---|---|-------------------------------|---|-------------------------------|---|-------------------------------|---|-------------------------------|---|-------------------------------|---|-------------------------------|---|------------------------------|---|------------------------------|----|------------------------------|----|-------------------------------|----|-------------------------------|----|-------------------------------|----|-------------------------------------|---|---|--|---|-------------------------------|---|-------------------------------|---|-------------------------------|---|------------------------------|---|------------------------------|---|-------------------------------|---|-------------------------------------|
| <table border="1"> <tbody> <tr><td>1</td><td><input checked="" type="checkbox"/> air</td></tr> <tr><td>2</td><td><input type="checkbox"/> pres</td></tr> <tr><td>3</td><td><input type="checkbox"/> rhum</td></tr> <tr><td>4</td><td><input type="checkbox"/> uwnd</td></tr> <tr><td>5</td><td><input type="checkbox"/> wwnd</td></tr> <tr><td>6</td><td><input type="checkbox"/> WDIR</td></tr> <tr><td>7</td><td><input type="checkbox"/> WSPD</td></tr> <tr><td>8</td><td><input type="checkbox"/> GST</td></tr> <tr><td>9</td><td><input type="checkbox"/> DPD</td></tr> <tr><td>10</td><td><input type="checkbox"/> APD</td></tr> <tr><td>11</td><td><input type="checkbox"/> PRES</td></tr> <tr><td>12</td><td><input type="checkbox"/> ATMP</td></tr> <tr><td>13</td><td><input type="checkbox"/> WTMP</td></tr> <tr><td>14</td><td><input type="checkbox"/> Class WVHT</td></tr> </tbody> </table> | 1  | <input checked="" type="checkbox"/> air | 2 | <input type="checkbox"/> pres | 3 | <input type="checkbox"/> rhum | 4 | <input type="checkbox"/> uwnd | 5 | <input type="checkbox"/> wwnd | 6 | <input type="checkbox"/> WDIR | 7 | <input type="checkbox"/> WSPD | 8 | <input type="checkbox"/> GST | 9 | <input type="checkbox"/> DPD | 10 | <input type="checkbox"/> APD | 11 | <input type="checkbox"/> PRES | 12 | <input type="checkbox"/> ATMP | 13 | <input type="checkbox"/> WTMP | 14 | <input type="checkbox"/> Class WVHT | <table border="1"> <tbody> <tr><td>1</td><td><input checked="" type="checkbox"/> pres</td></tr> <tr><td>2</td><td><input type="checkbox"/> rhum</td></tr> <tr><td>3</td><td><input type="checkbox"/> wwnd</td></tr> <tr><td>4</td><td><input type="checkbox"/> WDIR</td></tr> <tr><td>5</td><td><input type="checkbox"/> GST</td></tr> <tr><td>6</td><td><input type="checkbox"/> APD</td></tr> <tr><td>7</td><td><input type="checkbox"/> WTMP</td></tr> <tr><td>8</td><td><input type="checkbox"/> Class WVHT</td></tr> </tbody> </table> | 1 | <input checked="" type="checkbox"/> pres | 2 | <input type="checkbox"/> rhum | 3 | <input type="checkbox"/> wwnd | 4 | <input type="checkbox"/> WDIR | 5 | <input type="checkbox"/> GST | 6 | <input type="checkbox"/> APD | 7 | <input type="checkbox"/> WTMP | 8 | <input type="checkbox"/> Class WVHT |
| 1  | <input checked="" type="checkbox"/> air  |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 2  | <input type="checkbox"/> pres            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 3  | <input type="checkbox"/> rhum            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 4  | <input type="checkbox"/> uwnd            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 5  | <input type="checkbox"/> wwnd            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 6  | <input type="checkbox"/> WDIR            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 7  | <input type="checkbox"/> WSPD            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 8  | <input type="checkbox"/> GST             |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 9  | <input type="checkbox"/> DPD             |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 10   | <input type="checkbox"/> APD             |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 11   | <input type="checkbox"/> PRES            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 12   | <input type="checkbox"/> ATMP            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 13   | <input type="checkbox"/> WTMP            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 14   | <input type="checkbox"/> Class WVHT      |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 1  | <input checked="" type="checkbox"/> pres |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 2  | <input type="checkbox"/> rhum            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 3  | <input type="checkbox"/> wwnd            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 4  | <input type="checkbox"/> WDIR            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 5  | <input type="checkbox"/> GST             |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 6  | <input type="checkbox"/> APD             |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 7  | <input type="checkbox"/> WTMP            |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |
| 8  | <input type="checkbox"/> Class WVHT      |   |   |                               |   |                               |   |                               |   |                               |   |                               |   |                               |   |                              |   |                              |    |                              |    |                               |    |                               |    |                               |    |                                     |   |   |  |   |                               |   |                               |   |                               |   |                              |   |                              |   |                               |   |                                     |

Antes

Después

Figura 5.12: Eliminación de correlaciones.

A continuación, se mostrará el proceso seguido a la hora de eliminar los atributos correlacionados. El proceso se producirá iterativamente, es decir, primero se elimina el más correlacionado, tras ello, se vuelven a aplicar los filtros y se vuelven a tomar decisiones haciendo pruebas y extrayendo modelos, se elimina entonces el siguiente atributo. El resultado de eliminar todos los atributos se puede observar en la figura 5.12.

99% WSPD, GST ----> Se elimina WSPD (menos correlacionado con la clase)

98% air, ATMP -----> Se elimina air (menos correlacionado con la clase)

95% ATMP, WTMP -----> Se elimina ATMP (menos correlacionado con la clase)

93% pres, PRES -----> Se elimina PRES (menos correlacionado con la clase)

71% DPD, APD -----> Se elimina DPD (menos correlacionado con la clase)

68% uwnd, WDIR -----> Se elimina uwnd (menos correlacionado con la clase)

#### 5.1.5. Valores perdidos y outliers

Tras utilizar el filtro no supervisado a nivel de atributo de Weka de detección de *outliers* y valores extremos, *InterquartileRange*, con sus valores predeterminados no se detectó ningún outlier (Figura 5.13).

| Name: Outlier   |       | Type: Nominal  |
|-----------------|-------|----------------|
| Missing: 0 (0%) |       | Distinct: 1    |
|                 |       | Unique: 0 (0%) |
| No.             | Label | Count          |
| 1               | no    | 3559           |
| 2               | yes   | 0              |

Figura 5.13: Outlier detectados en la base de datos

### 5.1.6. Normalización

| 1: pres<br>Numeric | 2: rhum<br>Numeric | 3: vwind<br>Numeric | 4: WDIR<br>Numeric | 5: GST<br>Numeric | 6: APD<br>Numeric | 7: WTMP<br>Numeric |
|--------------------|--------------------|---------------------|--------------------|-------------------|-------------------|--------------------|
| 101320.0           | 92.0               | 2.699997            | 256.0              | 7.1               | 6.3...            | 286.55             |
| 101190.0           | 89.0               | 2.800003            | 223.0              | 4.9               | 5.16              | 286.55             |
| 101050.0           | 85.0               | 3.900009            | 206.0              | 3.8               | 5.23              | 286.95             |
| 100830.0           | 86.0               | 2.400009            | 127.0              | 2.0               | 5.84              | 286.95             |
| 100740.0           | 91.0               | -1.399994           | 13.0               | 3.4               | 5.83              | 286.85             |
| 100680.0           | 90.0               | -2.099991           | 25.0               | 2.8               | 6.4               | 286.85             |
| 100810.0           | 84.0               | -2.199997           | 348.0              | 1.5               | 6.22              | 288.05             |
| 100840.0           | 81.0               | -1.899994           | 331.0              | 5.6               | 5.94              | 287.15             |
| 100840.0           | 86.0               | -3.099991           | 352.0              | 4.3               | 5.84              | 287.05             |
| 100880.0           | 86.0               | -2.199997           | 320.0              | 5.5               | 5.64              | 287.05             |
| 100990.0           | 83.0               | 0.5                 | 286.0              | 6.6               | 5.01              | 287.15             |
| 101100.0           | 84.0               | 2.699997            | 254.0              | 5.9               | 4.89              | 287.15             |
| 101200.0           | 89.0               | 4.0                 | 246.0              | 5.0               | 4.95              | 287.05             |
| 101420.0           | 83.0               | 3.5                 | 204.0              | 3.7               | 5.24              | 287.05             |
| 101500.0           | 78.0               | 2.600006            | 155.0              | 2.8               | 5.4               | 287.75             |
| 101470.0           | 76.0               | 0.5                 | 87.0               | 2.5               | 5.72              | 287.85             |
| 101370.0           | 84.0               | -1.099991           | 31.0               | 5.9               | 5.47              | 287.35             |
| 101410.0           | 85.0               | -2.399994           | 4.0                | 6.1               | 4.07              | 287.35             |
| 101420.0           | 88.0               | -2.099991           | 332.0              | 7.5               | 3.7               | 287.55             |
| 101340.0           | 88.0               | -1.800003           | 318.0              | 7.3               | 4.08              | 287.55             |
| 101270.0           | 93.0               | 0.900009            | 272.0              | 5.9               | 3.84              | 287.55             |
| 101260.0           | 88.0               | 4.100006            | 194.0              | 5.2               | 4.78              | 287.45             |
| 101370.0           | 81.0               | 5.800003            | 241.0              | 10.8              | 3.38              | 287.65             |
| 101480.0           | 80.0               | 6.100006            | 217.0              | 11.2              | 4.19              | 287.35             |
| 101520.0           | 89.0               | 6.199997            | 181.0              | 5.4               | 4.19              | 287.25             |
| 101680.0           | 89.0               | 7.600006            | 152.0              | 8.6               | 4.4               | 287.25             |
| 101880.0           | 86.0               | 6.600006            | 120.0              | 9.7               | 4.57              | 287.35             |
| 102060.0           | 88.0               | 5.199997            | 119.0              | 9.9               | 4.69              | 287.25             |
| 102230.0           | 90.0               | 4.600006            | 109.0              | 8.8               | 4.94              | 287.25             |
| 102350.0           | 82.0               | 1.699997            | 111.0              | 8.1               | 4.63              | 287.25             |

Figura 5.14: Base de datos antes de normalizar.

| 1: pres<br>Numeric | 2: rhum<br>Numeric | 3: vwind<br>Numeric | 4: WDIR<br>Numeric | 5: GST<br>Numeric | 6: APD<br>Numeric | 7: WTMP<br>Numeric |
|--------------------|--------------------|---------------------|--------------------|-------------------|-------------------|--------------------|
| 0.676884           | 0.846154           | 0.528205            | 0.713092           | 0.2607            | 0.412041          | 0.732673           |
| 0.660281           | 0.788462           | 0.530769            | 0.62117            | 0.175097          | 0.257822          | 0.732673           |
| 0.642401           | 0.711538           | 0.558975            | 0.573816           | 0.132296          | 0.266583          | 0.772277           |
| 0.614304           | 0.730769           | 0.520513            | 0.35376            | 0.062257          | 0.342929          | 0.772277           |
| 0.60281            | 0.826923           | 0.423077            | 0.036212           | 0.116732          | 0.341677          | 0.762376           |
| 0.595147           | 0.807692           | 0.405128            | 0.069638           | 0.093385          | 0.413016          | 0.762376           |
| 0.61175            | 0.692308           | 0.402564            | 0.969359           | 0.042802          | 0.390488          | 0.881188           |
| 0.615581           | 0.634615           | 0.410257            | 0.922006           | 0.202335          | 0.355444          | 0.792079           |
| 0.615581           | 0.730769           | 0.379487            | 0.980501           | 0.151751          | 0.342929          | 0.782178           |
| 0.62069            | 0.730769           | 0.402564            | 0.891365           | 0.198444          | 0.317897          | 0.782178           |
| 0.634738           | 0.673077           | 0.471795            | 0.796657           | 0.241245          | 0.239049          | 0.792079           |
| 0.648787           | 0.692308           | 0.528205            | 0.707521           | 0.214008          | 0.22403           | 0.792079           |
| 0.661558           | 0.788462           | 0.561538            | 0.685237           | 0.178988          | 0.231539          | 0.782178           |
| 0.689655           | 0.673077           | 0.548718            | 0.568245           | 0.128405          | 0.267835          | 0.782178           |
| 0.699872           | 0.576923           | 0.525641            | 0.431755           | 0.093385          | 0.28786           | 0.851485           |
| 0.696041           | 0.538462           | 0.471795            | 0.24234            | 0.081712          | 0.32791           | 0.861386           |
| 0.683269           | 0.692308           | 0.430769            | 0.086351           | 0.214008          | 0.296621          | 0.811881           |
| 0.688378           | 0.711538           | 0.397436            | 0.011142           | 0.22179           | 0.121402          | 0.811881           |
| 0.689655           | 0.769231           | 0.405128            | 0.924791           | 0.276265          | 0.075094          | 0.831683           |
| 0.679438           | 0.769231           | 0.41282             | 0.885794           | 0.268482          | 0.122653          | 0.831683           |
| 0.670498           | 0.865385           | 0.482051            | 0.75766            | 0.214008          | 0.092616          | 0.831683           |
| 0.669221           | 0.769231           | 0.564103            | 0.54039            | 0.18677           | 0.210263          | 0.821782           |
| 0.683269           | 0.634615           | 0.607692            | 0.671309           | 0.404669          | 0.035044          | 0.841584           |
| 0.697318           | 0.615385           | 0.615385            | 0.604457           | 0.420233          | 0.136421          | 0.811881           |
| 0.702427           | 0.788462           | 0.617949            | 0.504178           | 0.194553          | 0.136421          | 0.80198            |
| 0.722861           | 0.788462           | 0.653846            | 0.423398           | 0.319066          | 0.162703          | 0.80198            |
| 0.748404           | 0.730769           | 0.628205            | 0.334262           | 0.361868          | 0.18398           | 0.811881           |
| 0.771392           | 0.769231           | 0.592308            | 0.331476           | 0.36965           | 0.198999          | 0.80198            |
| 0.793103           | 0.807692           | 0.576923            | 0.303621           | 0.326848          | 0.230288          | 0.80198            |
| 0.808429           | 0.653846           | 0.502564            | 0.309192           | 0.299611          | 0.191489          | 0.80198            |

Figura 5.15: Base de datos tras normalizar.

El siguiente paso que se va a realizar sobre el *dataset* será una transformación lineal en el intervalo [0,1]. El objetivo de esto es conseguir que todos los atributos tengan la misma importancia, a la vez que, se conservan las relaciones entre ellos. En el caso de la normalización del conjunto de *train*, se hará haciendo uso del filtro no supervisado que proporciona *weka* para atributos, denominado *Normalize*. En la figura 5.14, se observa como se encontraban los datos antes de su normalización, mientras que el cambio se ve reflejado en la figura 5.15.

Para llevar a cabo el proceso de normalización en el conjunto de *test* se hará en función del valor mínimo y máximo que tiene cada atributo en el conjunto de *train*. Para facilitar la tarea se ha utilizado el filtro no supervisado a nivel de atributo *MathExpression*.

### 5.1.7. Selección de características

Con este proceso lo que se persigue es la obtención de una representación reducida del conjunto de datos que preserve la información relevante del conjunto original. Para esto se han empleado un método *Filter* y un método *Wrapper* con el algoritmo *MultilayerPerceptron*.

De los atributos que indican ambos métodos como sobrantes se ha eliminado el atributo *rhum* ya que coincidía en ambos casos.

Para el algoritmo *Filter* se ha empleado el evaluador de atributos *CfsSubsetEval* con los métodos de búsqueda *BestFirst* y *GreedyStepwise*.

Para el algoritmo *Wrapper* se ha empleado el método de búsqueda *BestFirst*.

| Antes  | Después  |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
|--|--|--|--|--|--|--|--|---|---|---|---|--|--|--|--|--|---|--|--|--|--|--|---|---|---|---|--|--|--|--|--|
| <table><tbody><tr><td>1 <input checked="" type="checkbox"/> pres</td><td>1 <input checked="" type="checkbox"/> pres</td></tr><tr><td>2 <input checked="" type="checkbox"/> rhum</td><td>2 <input checked="" type="checkbox"/> vwnd</td></tr><tr><td>3 <input checked="" type="checkbox"/> vwnd</td><td>3 <input checked="" type="checkbox"/> WDIR</td></tr><tr><td>4 <input checked="" type="checkbox"/> WDIR</td><td>4 <input checked="" type="checkbox"/> GST</td></tr><tr><td>5 <input checked="" type="checkbox"/> GST</td><td>5 <input checked="" type="checkbox"/> APD</td></tr><tr><td>6 <input checked="" type="checkbox"/> APD</td><td>6 <input checked="" type="checkbox"/> WTMP</td></tr><tr><td>7 <input checked="" type="checkbox"/> WTMP</td><td>7 <input checked="" type="checkbox"/> Class_WVHT</td></tr><tr><td>8 <input checked="" type="checkbox"/> Class_WVHT</td><td></td></tr></tbody></table> | 1 <input checked="" type="checkbox"/> pres       | 1 <input checked="" type="checkbox"/> pres | 2 <input checked="" type="checkbox"/> rhum | 2 <input checked="" type="checkbox"/> vwnd | 3 <input checked="" type="checkbox"/> vwnd | 3 <input checked="" type="checkbox"/> WDIR | 4 <input checked="" type="checkbox"/> WDIR | 4 <input checked="" type="checkbox"/> GST | 5 <input checked="" type="checkbox"/> GST | 5 <input checked="" type="checkbox"/> APD | 6 <input checked="" type="checkbox"/> APD | 6 <input checked="" type="checkbox"/> WTMP | 7 <input checked="" type="checkbox"/> WTMP | 7 <input checked="" type="checkbox"/> Class_WVHT | 8 <input checked="" type="checkbox"/> Class_WVHT |  | <table><tbody><tr><td>1 <input checked="" type="checkbox"/> pres</td><td>1 <input checked="" type="checkbox"/> pres</td></tr><tr><td>2 <input checked="" type="checkbox"/> vwnd</td><td>2 <input checked="" type="checkbox"/> WDIR</td></tr><tr><td>3 <input checked="" type="checkbox"/> WDIR</td><td>3 <input checked="" type="checkbox"/> GST</td></tr><tr><td>4 <input checked="" type="checkbox"/> GST</td><td>4 <input checked="" type="checkbox"/> APD</td></tr><tr><td>5 <input checked="" type="checkbox"/> APD</td><td>5 <input checked="" type="checkbox"/> WTMP</td></tr><tr><td>6 <input checked="" type="checkbox"/> WTMP</td><td>6 <input checked="" type="checkbox"/> Class_WVHT</td></tr><tr><td>7 <input checked="" type="checkbox"/> Class_WVHT</td><td></td></tr></tbody></table> | 1 <input checked="" type="checkbox"/> pres | 1 <input checked="" type="checkbox"/> pres | 2 <input checked="" type="checkbox"/> vwnd | 2 <input checked="" type="checkbox"/> WDIR | 3 <input checked="" type="checkbox"/> WDIR | 3 <input checked="" type="checkbox"/> GST | 4 <input checked="" type="checkbox"/> GST | 4 <input checked="" type="checkbox"/> APD | 5 <input checked="" type="checkbox"/> APD | 5 <input checked="" type="checkbox"/> WTMP | 6 <input checked="" type="checkbox"/> WTMP | 6 <input checked="" type="checkbox"/> Class_WVHT | 7 <input checked="" type="checkbox"/> Class_WVHT |  |
| 1 <input checked="" type="checkbox"/> pres   | 1 <input checked="" type="checkbox"/> pres       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 2 <input checked="" type="checkbox"/> rhum   | 2 <input checked="" type="checkbox"/> vwnd       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 3 <input checked="" type="checkbox"/> vwnd   | 3 <input checked="" type="checkbox"/> WDIR       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 4 <input checked="" type="checkbox"/> WDIR   | 4 <input checked="" type="checkbox"/> GST        |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 5 <input checked="" type="checkbox"/> GST  | 5 <input checked="" type="checkbox"/> APD        |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 6 <input checked="" type="checkbox"/> APD  | 6 <input checked="" type="checkbox"/> WTMP       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 7 <input checked="" type="checkbox"/> WTMP   | 7 <input checked="" type="checkbox"/> Class_WVHT |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 8 <input checked="" type="checkbox"/> Class_WVHT   |  |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 1 <input checked="" type="checkbox"/> pres   | 1 <input checked="" type="checkbox"/> pres       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 2 <input checked="" type="checkbox"/> vwnd   | 2 <input checked="" type="checkbox"/> WDIR       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 3 <input checked="" type="checkbox"/> WDIR   | 3 <input checked="" type="checkbox"/> GST        |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 4 <input checked="" type="checkbox"/> GST  | 4 <input checked="" type="checkbox"/> APD        |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 5 <input checked="" type="checkbox"/> APD  | 5 <input checked="" type="checkbox"/> WTMP       |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 6 <input checked="" type="checkbox"/> WTMP   | 6 <input checked="" type="checkbox"/> Class_WVHT |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |
| 7 <input checked="" type="checkbox"/> Class_WVHT   |  |  |  |  |  |  |  |   |   |   |   |  |  |  |  |  |   |  |  |  |  |  |   |   |   |   |  |  |  |  |  |

Figura 5.16: Eliminación de atributos tras la selección de características.

### 5.1.8. Búsqueda del mejor algoritmo y parámetros

Tras todo el tratamiento comentado anteriormente, se comenzó a entrenar modelos y a evaluarlos con un conjunto de validación, que representaba un año completo de mediciones, y con un *10Fold*.

|                                    |                                       |
|------------------------------------|---------------------------------------|
| bagSizePercent                     | 100                                   |
| batchSize                          | 100                                   |
| calcOutOfBag                       | False                                 |
| classifier                         | Choose MultilayerPerceptron -L 0.3 -N |
| debug                              | False                                 |
| doNotCheckCapabilities             | False                                 |
| numDecimalPlaces                   | 2                                     |
| numExecutionSlots                  | 1                                     |
| numIterations                      | 10                                    |
| outputOutOfBagComplexityStatistics | False                                 |
| printClassifiers                   | False                                 |
| representCopiesUsingWeights        | False                                 |
| seed                               | 1                                     |
| storeOutOfBagPredictions           | False                                 |

Figura 5.17: Parámetros del clasificador Bagging.

|                        |       |
|------------------------|-------|
| GUI                    | False |
| autoBuild              | True  |
| batchSize              | 100   |
| debug                  | False |
| decay                  | False |
| doNotCheckCapabilities | False |
| hiddenLayers           | 17    |
| learningRate           | 0.3   |
| momentum               | 0.2   |
| nominalToBinaryFilter  | True  |
| normalizeAttributes    | False |
| normalizeNumericClass  | True  |
| numDecimalPlaces       | 2     |
| reset                  | True  |
| seed                   | 0     |
| trainingTime           | 400   |
| validationSetSize      | 0     |

Figura 5.18: Parámetros del clasificador MultilayerPerceptron utilizado por Bagging

Se construyeron diversos modelos, pero el que mejor puntuación obtenía era un clasificador de tipo *Meta*, denominado *Bagging*. En la *figura 5.17*, se pueden observar sus parámetros, así como que uno de sus parámetros es otro clasificador. Para este otro clasificador se empleo el *MultilayerPerceptron* de *Weka*, con los parámetros que se indican en la *figura 5.18*.

|               |  |         |    |
|---------------|--|---------|----|
| IAA1819Grupo7 |  | 0.74350 | 84 |
|---------------|--|---------|----|

*Figura 5.19: Resultados tras aplicar el clasificador.*

El resultado obtenido al subir los resultados de clasificar el test con este modelo, se pueden observar en la *figura 5.19*, donde el *FMeasure* es 0.74350 y las subidas totales de resultados son 84.