Práctica 3: Integración de *KNIME* y *WEKA*

Sistemas de Ayuda a la Toma de Decisiones

Pedro Allué Tamargo (758267) Juan José Tambo Tambo (755742)
Jesús Villacampa Sagaste (755739)

4 de noviembre de 2020

Índice

1. Ejercicio 1

Se ha creado el *workflow* ilustrado en la Figura 1 para trabajar con los datos del conjunto de datos *yellow-small.data*.
Se ha utilizado un nodo *Rule Engine* para crear una nueva columna *"class"*. El contenido de este nodo son las siguientes reglas:

```
$Color$ MATCHES "YELLOW" AND $Size$ MATCHES "SMALL" => "inflated"
TRUE => "not inflated"
```

Tras este nodo se ha utilizado un nodo *String manipulation* para concatenar los valores de las columnas *class* e *inflated (true/false)* utilizando la expresión:

```
string($class$ + " is " + $Inflated (True/False)$)
```

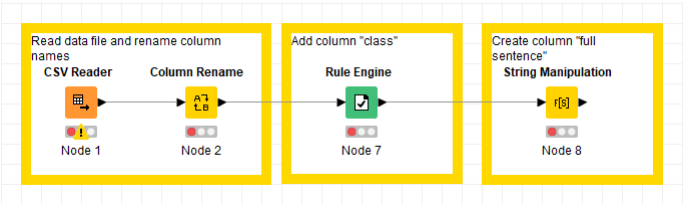

Figura 1: Workflow del ejercicio 1

2. Ejercicio 2

Se va a proceder a analizar un conjunto de datos que describe el número de visitantes de un sitio web en los meses de junio/julio de 2010 (fichero *website1.txt*).

Para calcular los parámetros de media, desviación típica, Kurtosis se ha utilizado el *workflow* mostrado en la Figura 2.
La *Kurtosis* es una medida estadística que muestra la forma de una distribución de probabilidad. Una *Kurtosis* grande implica una mayor concentración de valores de la variables o muy cerca de la media de la distribución (pico) o muy lejos de ella (colas de la distribución), al mismo tiempo que existe una menor frecuencia de valores intermedios.
Una *Kurtosis* pequeña (como en este caso cuyo valor es 0.466) implica que los datos se establecen alrededor de la media, es decir, que el conjunto de datos no presenta *datos atípicos*.

Para entrenar la red Bayesiana (Figura 2) se deben preparar los datos. Para ello se debe crear una nueva columna *isWeekend* para ilustrar si es fin de semana o no. Utilizando el nodo *Rule engine* se utilizarán las siguientes reglas:

```
$weekday$ MATCHES "Sat" => "Yes"
$weekday$ MATCHES "Sun" => "Yes"
TRUE => "No"
```

Se ha utilizado un nodo *Column filter* para eliminar la columna *weekday* ya que la red Bayesiana presenta un mejor rendimiento si conoce este valor ya que si se entrena con esta variable reconoce la regla de creación de la columna *isWeekend*.
Para llegar a esta conclusión se han probado las distintas combinaciones de columnas utilizando el *Column Filter*, eligiendo las columnas que se utilizarán para entrenar la red *Bayesiana* y comparando con cual de ellas se obtiene un mejor rendimiento.

Para dibujar la curva *ROC* se ha utilizado un nodo *ROC Curve* a la salida del nodo *Naive Bayes Predictor*. Se puede observar en la Figura 3 que la gráfica se encuentra mayoritariamente en la sección superior del triángulo. Esto indica que el modelo es capaz de identificar correctamente los verdaderos positivos. La curva *ROC* se utiliza para estudiar la sensibilidad y la especialidad de un test diagnóstico.

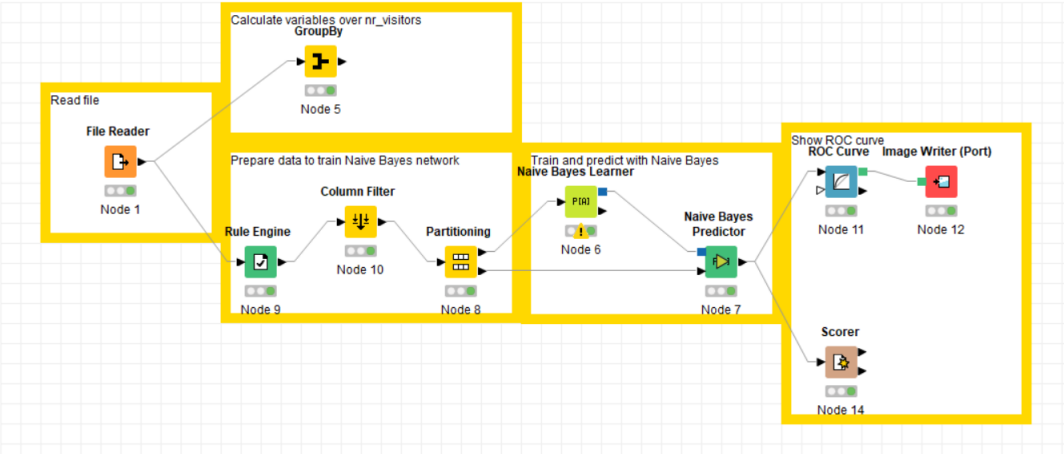

Figura 2: Workflow de ejercicio 2

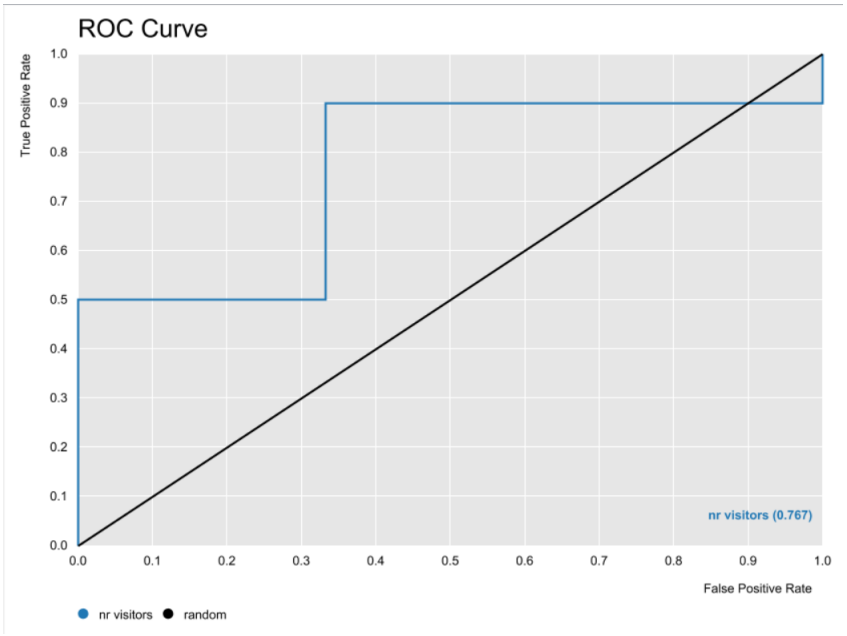

Figura 3: Curva ROC del ejercicio 2

3. Ejercicio 3

Para la evaluación de los distintos conjuntos de datos se han creado 2 *workflows*. Uno de ellos (Figura 6) utiliza las herramientas de *KNIME* para evaluar los datos. El otro (Figura 5) utiliza las herramientas de *WEKA* para evaluar los datos.
En el anexo se pueden observar las tablas comparativas de las distintas ejecuciones de los *workflows* tanto con *KNIME* (K) como con *WEKA* (W). Para el conjunto de datos *wine* se han entrenado las distintas herramientas con el 80 %, 50 % y 30 %.

Para el conjunto de datos *wine* se han entrenado las distintas herramientas con el 80 %, 50 % y 30 %. Se puede apreciar que conforme se reduce el número de datos de entrenamiento se pierde precisión, aunque de de una manera muy ligera ya que no es un conjunto grande de datos ni se ha ejecutado una cantidad grande de veces para que se estabilicen los resultados. El método que mejor ha funcionado identificando de las clases ha sido el perceptrón multicapa, ya que se ha utilizado la configuración del perceptrón que mejor funcionaba en la anterior práctica.
Cabe destacar que tanto el perceptrón multicapa como Naive Bayes realizan una clasificación del tipo 3 muy correcta, pero J48 erra bastante con respecto a los dos anteriores, prácticamente siempre por debajo del 0.9 de TP.

Para el conjunto de datos *iris* se han entrenado las distintas herramientas con el 80 %, 50 % y 30 % de los datos.

Se puede observar que en todos los modelos se obtienen los resultados más precisos cuando se usa un mayor número de datos de entrenamiento (80 %), llegando en algunos casos a clasificar correctamente el 100 % de los elementos pertenecientes a la clase *Iris-Setosa*, ya que la tasa de *True Positives* es 1.

Los peores resultados se han obtenido con el modelo de *perteptrón multicapa*, sobre todo con los nodos de *Knime*, ya que la media de *True Positives* ronda el 80 % para cada una de las clases. La mayor tasa de *True Positives* se ha obtenido con el modelo *J48*, sobre todo con los nodos de *Knime*, ya que se obtiene una media de *True Positives* superior al 90 % para cada una de las clases, llegando incluso al 100 % para la clase *Iris-Setosa*. El modelo *Naive Bayes* consigue unos resultados muy parecidos al de *J48*, sobre todo con los nodos de *Weka*.

Para el conjunto de datos *adult* se han entrenado las distintas herramientas con el 80 %, 50 % y 30 %. Se puede apreciar que conforme se reduce el número de datos de entrenamiento se pierde precisión, identificando datos pertenecientes a la otra clase de datos (>50K) como datos de clase <=50K. El método que más ha errado en la identificación de las clases ha sido el perceptrón multicapa.
El método que menos ha variado la precisión conforme se entrenaba con menos datos ha sido el *J48* y en especial utilizando los nodos de *WEKA* siempre ha mantenido el porcentaje de *True Positives* por encima del 93 % en la clase <=50K y del 60 % en la clase >50K.

Se ha planteado una gráfica (Figura 4) para el conjunto de datos *Iris* en la cual se compara la precisión de los distintos métodos conforme se van reduciendo los datos de entrenamiento. Se puede observar un descenso de la precisión en las clases *Iris-Versicolor* e *Iris-Virgínica* conforme se reducen los datos. Se puede observar como el perceptrón no varía especialmente conforme se reducen los datos de entrenamiento. Este fenómeno puede ocurrir debido a que el particionado de los datos se ha realizado de forma aleatoria con el nodo *partitioning* y se han realizado 5 ejecuciones para hallar una media en la precisión de la identificación de las clases.

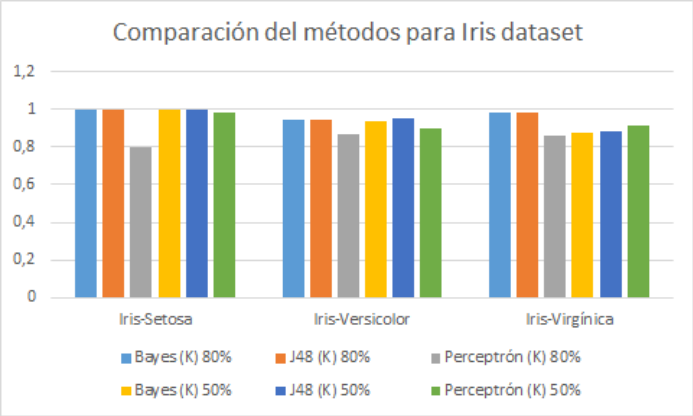

Figura 4: Gráfica comparativa de los métodos para el dataset *Iris*

En cuanto a los problemas que han surgido, con el *dataset adult* se necesitaba una columna *Class* y por lo tanto, se ha renombrado la última columna a *"Class"*. Para ello se ha utilizado un nodo *Column Filter* para eliminar todas las variables no numéricas.

Para simplificar y agilizar el entrenamiento y ejecución de cada uno de los clasificadores, se han creado dos *workflows*, uno que agrupa todos los clasificadores que provee la propia herramienta *Knime* (Figura 6) y otro que agrupa los nodos que proporciona el *plugin* de *Weka* (Figura 5).

De esta manera en cada *Workflow* se indica el *dataset* deseado con el nodo *File Reader* y se selecciona el porcentaje de datos a entrenar con el nodo *partitioning*. Los resultados se escriben en archivos *csv* distintos, indicados con los nodos *CSV Writer*. Así se centralizan los diferentes modelos de entrenamiento con una sola ejecución.

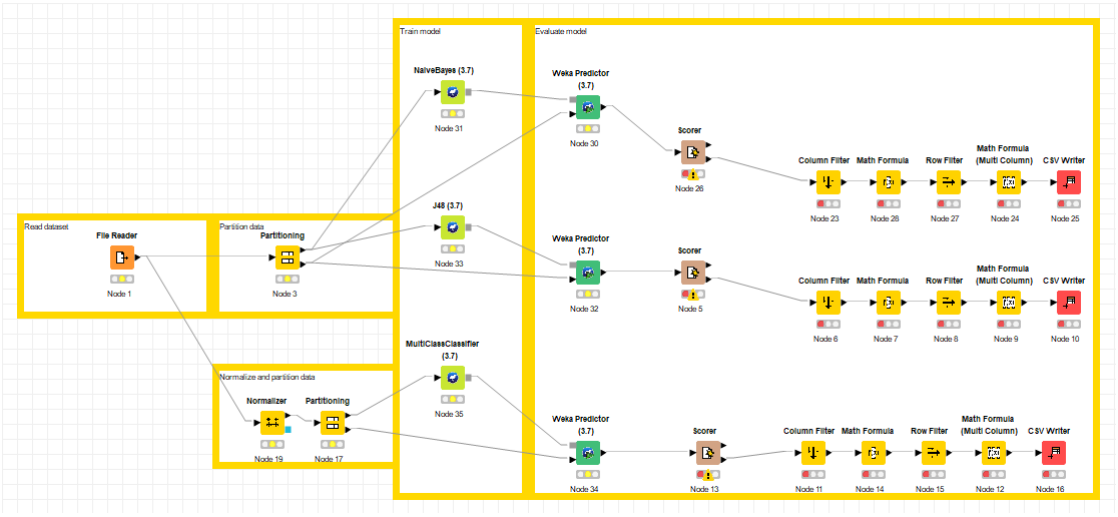

Figura 5: Workflow con modelos de entrenamiento de *Weka*

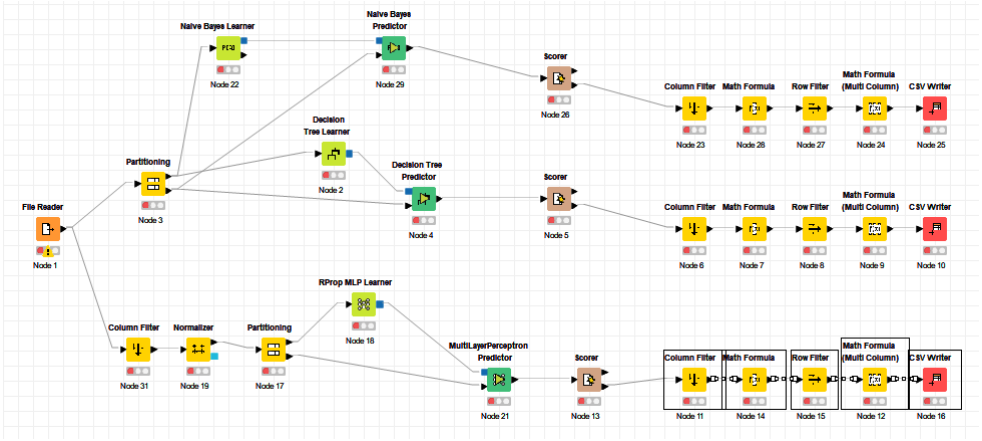

Figura 6: Workflow con modelos de entrenamiento de *Knime*

4. Anexo 1: Cuadros de datos ejercicio 3

4.1. Dataset Adult

4.1.1. Naive Bayes

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.942	0.058	0.936	0.064
>50K	0.511	0.489	0.515	0.485

Cuadro 1: Datos de entrenamiento del dataset *Adult* con el 80 % usando *Naive Bayes*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.941	0.059	0.933	0.067
>50K	0.513	0.487	0.513	0.487

Cuadro 2: Datos de entrenamiento del dataset *Adult* con el 50 % usando *Naive Bayes*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.939	0.061	0.934	0.066
>50K	0.508	0.492	0.522	0.478

Cuadro 3: Datos de entrenamiento del dataset *Adult* con el 30 % usando *Naive Bayes*

4.1.2. J48

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.900	0.100	0.936	0.064
>50K	0.603	0.397	0.621	0.379

Cuadro 4: Datos de entrenamiento del dataset *Adult* con el 80 % usando *J48*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.897	0.103	0.937	0.063
>50K	0.598	0.402	0.607	0.392

Cuadro 5: Datos de entrenamiento del dataset *Adult* con el 50 % usando *J48*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.890	0.110	0.933	0.067
>50K	0.599	0.401	0.600	0.400

Cuadro 6: Datos de entrenamiento del dataset *Adult* con el 30 % usando *J48*

4.1.3. Perceptrón multicapa

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.901	0.099	0.927	0.073
>50K	0.377	0.623	0.602	0.398

Cuadro 7: Datos de entrenamiento del dataset *Adult* con el 80 % usando *Perceptrón multicapa*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	1.000	0.000	0.930	0.070
>50K	0.000	1.000	0.605	0.395

Cuadro 8: Datos de entrenamiento del dataset *Adult* con el 50 % usando *Perceptrón multicapa*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
<=50K	0.890	0.110	0.932	0.068
>50K	0.389	0.611	0.587	0.413

Cuadro 9: Datos de entrenamiento del dataset *Adult* con el 30 % usando *Perceptrón multicapa*

4.2. Dataset Iris

4.2.1. Naive Bayes

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	1	0	1	0
Iris-versicolor	0,947	0,053	0,928	0,072
Iris-virginica	0,983	0,017	0,913	0,087

Cuadro 10: Datos de entrenamiento del dataset *Iris* con el 80 % usando *Naive Bayes*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	1	0	1	0
Iris-versicolor	0,938	0,062	0,895	0,105
Iris-virginica	0,874	0,126	0,912	0,088

Cuadro 11: Datos de entrenamiento del dataset *Iris* con el 50 % usando *Naive Bayes*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	1	0	1	0
Iris-versicolor	0,904	0,096	0,814	0,186
Iris-virginica	0,883	0,117	0,932	0,068

Cuadro 12: Datos de entrenamiento del dataset *Iris* con el 20 % usando *Naive Bayes*

4.2.2. J48

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	1	0	1	0
Iris-versicolor	0,947	0,053	0,903	0,097
Iris-virginica	0,983	0,017	0,910	0,090

Cuadro 13: Datos de entrenamiento del dataset *Iris* con el 80 % usando *J48*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	1	0	0,968	0,032
Iris-versicolor	0,952	0,048	0,909	0,091
Iris-virginica	0,886	0,114	0,885	0,115

Cuadro 14: Datos de entrenamiento del dataset *Iris* con el 50 % usando *J48*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	1	0	0,922	0,078
Iris-versicolor	0,935	0,065	0,910	0,090
Iris-virginica	0,907	0,093	0,942	0,058

Cuadro 15: Datos de entrenamiento del dataset *Iris* con el 20 % usando *J48*

4.2.3. Perceptrón multicapa

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	0,8	0,2	1	0
Iris-versicolor	0,865	0,135	0,883	0,117
Iris-virginica	0,860	0,140	0,822	0,178

Cuadro 16: Datos de entrenamiento del dataset *Iris* con el 80 % usando *Perceptrón multicapa*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	0,984	0,016	1	0
Iris-versicolor	0,898	0,102	0,933	0,066
Iris-virginica	0,915	0,085	0,947	0,053

Cuadro 17: Datos de entrenamiento del dataset *Iris* con el 50 % usando *Perceptrón multicapa*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
Iris-Setosa	0,995	0,005	0,995	0,005
Iris-versicolor	0,797	0,203	0,885	0,115
Iris-virginica	0,853	0,147	0,832	0,168

Cuadro 18: Datos de entrenamiento del dataset *Iris* con el 20 % usando *Perceptrón multicapa*

4.3. Dataset Wine

4.3.1. Naive Bayes

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,968	0,031	0,966	0,033
2	0,959	0,040	0,939	0,060
3	0,984	0,015	1	0

Cuadro 19: Datos de entrenamiento del dataset *Wine* con el 80 % usando *Naive Bayes*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,961	0,038	0,956	0,043
2	0,953	0,056	0,943	0,105
3	1	0	1	1

Cuadro 20: Datos de entrenamiento del dataset *Wine* con el 50 % usando *Naive Bayes*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,955	0,044	0,978	0,021
2	0,946	0,053	0,944	0,055
3	0,994	0,005	0,983	0,016

Cuadro 21: Datos de entrenamiento del dataset *Wine* con el 30 % usando *Naive Bayes*

4.3.2. J48

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,941	0,058	0,984	0,015
2	0,887	0,112	0,923	0,076
3	0,860	0,139	0,969	0,030

Cuadro 22: Datos de entrenamiento del dataset *Wine* con el 80 % usando *J48*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,944	0,055	0,906	0,093
2	0,880	0,119	0,886	0,113
3	0,849	0,150	0,825	0,174

Cuadro 23: Datos de entrenamiento del dataset *Wine* con el 50 % usando *J48*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,846	0,153	0,949	0,050
2	0,867	0,132	0,864	0,135
3	0,858	0,141	0,871	0,128

Cuadro 24: Datos de entrenamiento del dataset *Wine* con el 30 % usando *J48*

4.3.3. Perceptrón multicapa

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,966	0,033	0,986	0,013
2	0,913	0,086	0,967	0,032
3	1	0	0,96	0,04

Cuadro 25: Datos de entrenamiento del dataset *Wine* con el 80 % usando *Perceptrón multicapa*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,992	0,007	1	0
2	0,950	0,049	0,989	0,010
3	0,959	0,040	0,939	0,060

Cuadro 26: Datos de entrenamiento del dataset *Wine* con el 50 % usando *Perceptrón multicapa*

Clase	True Positives (K)	False Positives (K)	True Positives (W)	False Positives (W)
1	0,980	0,019	1	0
2	0,928	0,071	0,956	0,043
3	0,977	0,022	1	0

Cuadro 27: Datos de entrenamiento del dataset *Wine* con el 30 % usando *Perceptrón multicapa*