

# **16 Junio 2020. Text Mining Clasificación**

## **Herramienta: Knime. Parte 2**

**Presentar un tutorial en que desarrolle la resolución de la tarea**

**Herramienta: Knime**

### **Parte A**

**El siguiente workflow puede ser útil para guiar el desarrollo de la tarea.**

- 1. Utilizando un nodo IO-File Reader cargar el archivo textt\_mining\_clas1.txt (puede ser necesario eliminar algún atributo)**
- 2. Mediante un nodo Partitioning subdivida los datos al azar dejando el 70% de los datos en la primera partición.**
- 3. Utilizando un nodo Decision Tree Learner (sin cambiar las opciones por defecto) utilizando el 70% de los datos de la primera partición genere un árbol de clasificación.**
- 4. ¿Cuál es el atributo que más discrimina?**
- 5. Agregue un nodo Decision Tree Predictor, para aplicar el árbol de decisión al 30% de los datos de testing.**
- 6. Mediante un nodo Scorer genere una matriz de confusión. Interprete la matriz de confusión generada.**

### **Parte B**

**En base al archivo text\_mining\_clas2.txt desarrolle la clasificación de la misma forma que se desarrolló en la parte A.**

## **Parte A**

Vamos a trabajar con la herramienta KNIME. La abrimos y creamos un nuevo espacio de trabajo, dejando las opciones por defecto.

Una vez creado nuestro espacio de trabajo, buscamos en “Node Repository” un nodo capaz de leer el archivo con el que vamos a trabajar. Por tanto elegimos un nodo “File Reader” y lo arrastramos a nuestro espacio de trabajo.

Con el nodo seleccionado, pulsamos F6 para configurarlo. Cargamos el archivo text\_mining\_clas1.txt y comprobamos que se ha leído correctamente el archivo.

Dialog - 2:1 - File Reader

File

Settings | Flow Variables | Memory Policy

Enter ASCII data file location: (press 'Enter' to update preview)

C:\Users\stief\Downloads\text\_mining\_clas1.txt

Browse...

☐ Preserve user settings for new location

Rescan

Basic Settings

☒ read row IDs

☒ read column headers

Column delimiter: <tab>

☒ ignore spaces and tabs

☐ Java-style comments

Advanced...

Single line comment:

Preview

Click column header to change column properties (\* = name/type user settings)

Row ID	I termino1	I termino2	I termino3	S Tipo
documen...	2	2	11	A
documen...	2	0	12	A
documen...	2	0	10	A
documen...	1	0	8	A
documen...	2	0	7	A
documen...	2	0	11	A
documen...	3	1	2	A
documen...	2	2	8	A
documen...	3	1	13	A
documen...	3	2	1	A
documen...	1	0	0	A
documen...	3	0	2	A
documen...	3	2	15	A
documen...	2	2	2	A
documen...	2	1	10	A
documen...	3	0	9	A
documen...	3	2	9	A
documen...	1	0	0	A
documen...	2	0	0	A
documen...	1	2	1	A
documen...	2	2	5	A
documen...	2	1	10	A
documen...	2	0	12	A
documen...	3	2	2	A
documen...	2	2	5	A

OK Apply Cancel ?

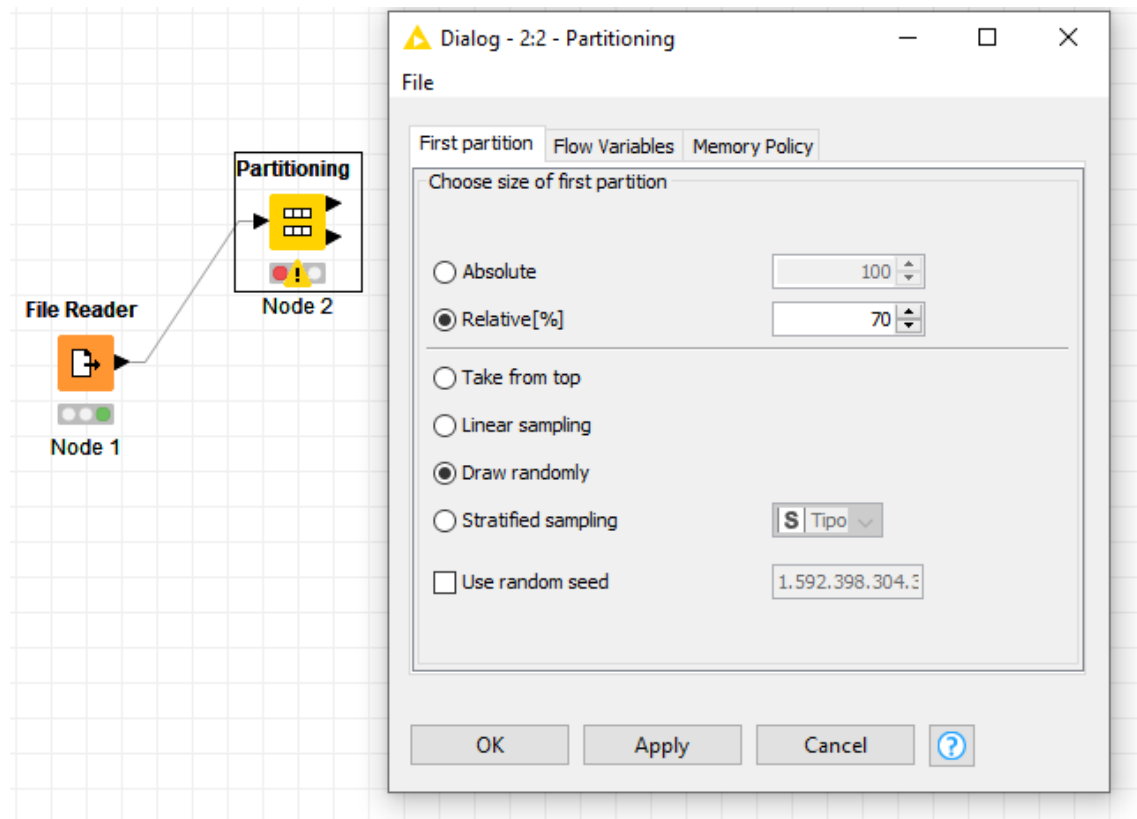
No es necesario eliminar la primera columna (Row ID) porque se identifica como una etiqueta directamente.

Nos podemos fijar en que si ponemos el puntero del ratón (mouse) en el triángulo que hay a la salida del nodo, que pone solamente que tiene 4 columnas la salida.

Vamos a dividir ahora los datos, para tener una parte de ellos para crear un modelo, y otra parte para testarlo.

Por tanto, buscamos el nodo “Partitioning” y lo arrastramos al espacio de trabajo, pulsando F6, una vez situado, para configurarlo.

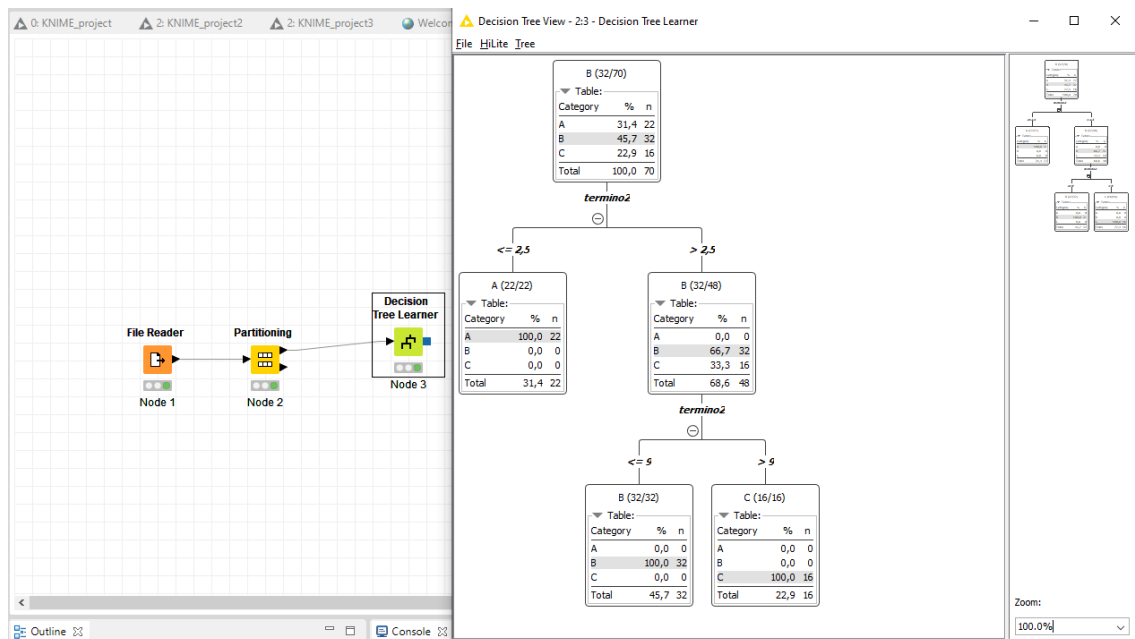
Aquí seleccionaremos que queremos los datos divididos en 70%-30%, y que queremos que estos datos se elijan al azar.



Pulsamos ok, y F7 para ejecutarlo.

Ahora vamos a buscar el nodo Decision Tree Learner y lo vamos a arrastrar al espacio de trabajo. Con este nodo vamos a crear el modelo, por lo que tiene que ir conectado a la pata del nodo Partitioning que presenta el 70% de los datos. Con los parámetros por defecto en la configuración, lo ejecutamos.

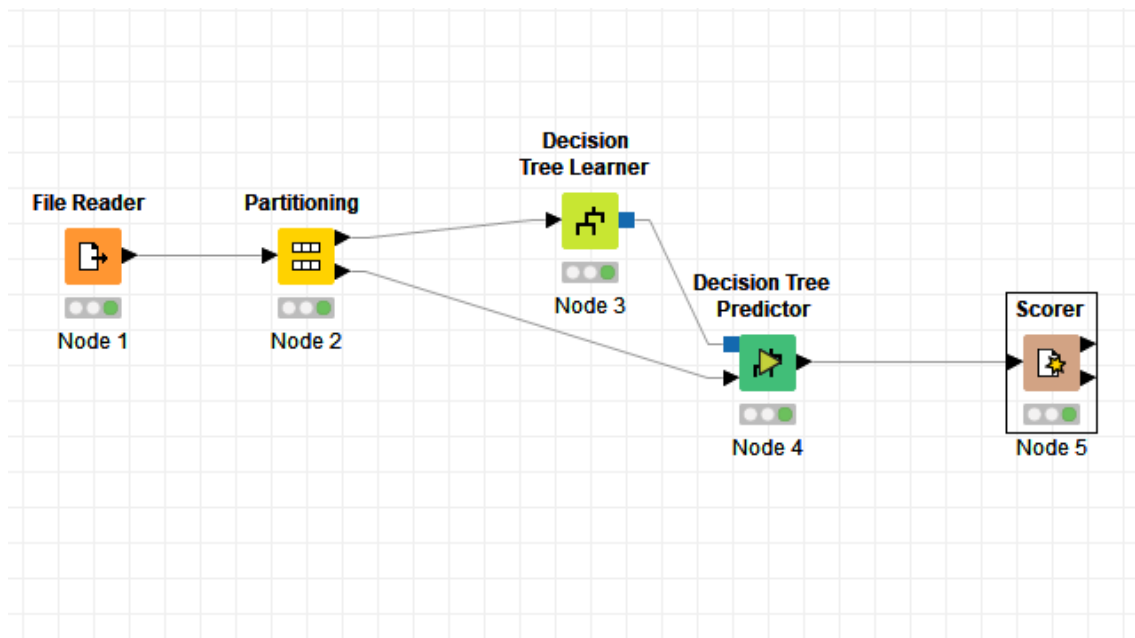
Observamos que se ha generado un árbol de decisión.



Vemos que el termino2 es el atributo que más discrimina.


Procedemos ahora a testear el modelo. Para ello, tendremos que usar un nodo Decision Tree Predictor con el 30% restante de los datos para testear el modelo, y luego visualizaremos con un nodo Scorer, la matriz de confusión.

Añadimos los nodos que acabamos de decir al espacio de trabajo y los conectamos de la siguiente manera:



Los ejecutamos dejando los parámetros por defecto de configuración.

Haciendo click derecho en el nodo Scorer podremos ver la matriz de confusión:

 Confusion Matrix - 2:5 - Scorer			
File Hilite			
Tipo \ Pred...	A	B	C
A	8	1	0
B	0	15	0
C	0	1	5

Como vemos en la diagonal principal, la mayor parte se han clasificado correctamente.

Sin embargo, un elemento que era tipo A se clasificó como tipo B erróneamente, y un elemento que era tipo C se clasificó erróneamente como de tipo B.

## **PARTE B**

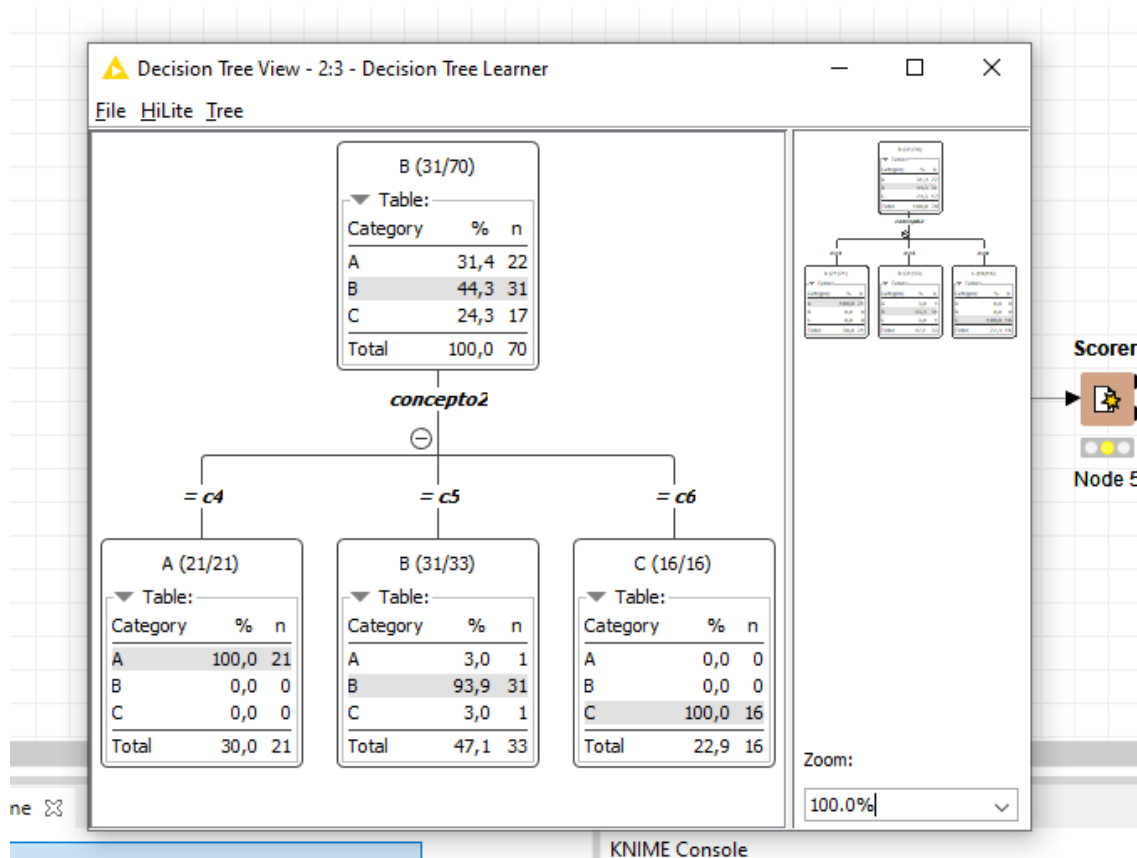
En esta parte lo que vamos a cambiar es con los datos que vamos a trabajar, pero los nodos seguirán siendo los mismos, por lo que trabajaremos sobre el espacio de trabajo creado en la parte A.

Primero vamos al nodo File Reader, y pulsamos F6 para configurarlo. Seleccionamos el nuevo archivo a tratar, y comprobamos que se abre correctamente:



Pulsamos Ok y F7 para ejecutarlo.

Ahora nos vamos al nodo Decision Tree Learner, y haciendo lo dejamos con la configuración por defecto y lo ejecutamos. Aquí podremos observar el árbol de decisión creado:



Vemos que el atributo que más discrimina es concepto2.

Ahora, seguimos ejecutando los nodos con la configuración por defecto hasta ejecutar el nodo Scorer. Aquí podremos visualizar la matriz de confusión:

**Confusion matrix - 2:5 - Scorer**

File Hilite Navigation View

Table "spec\_name" - Rows: 3 Spec - Columns: 3 Properties Flow Variables

Row ID	A	B	C
A	9	0	0
B	0	16	0
C	0	0	5

Como vemos, se han clasificado bien todos los elementos.