



Preparación de datos con Knime

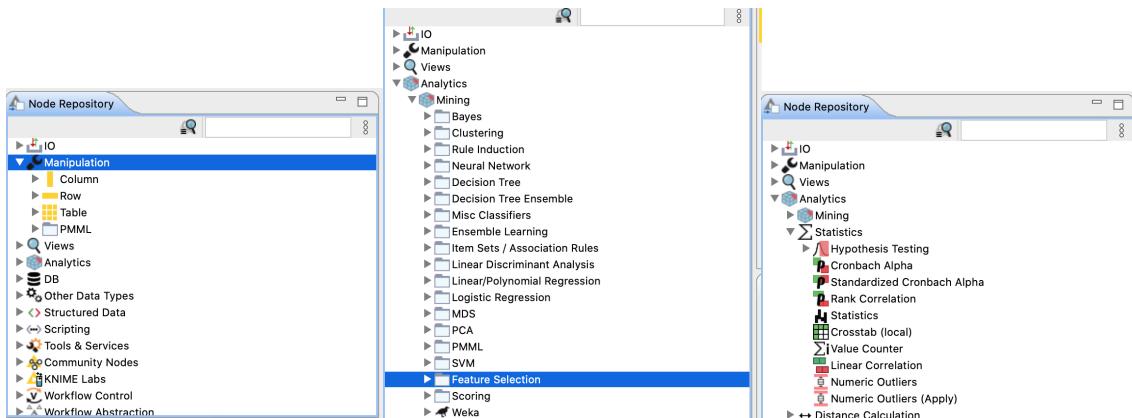
Tratamiento Inteligente de Datos
Master Universitario en Ingeniería Informática

1. Objetivos

En esta práctica veremos la importancia de la preparación de los datos en el proceso de extracción de patrones. Se trabajará con un conjunto de datos reales (un subconjunto, en realidad) sobre el que se aplicarán distintas técnicas de preparación de datos tales como la imputación o selección de características e instancias. Para valorar la utilidad de las distintas técnicas, se entrenará un modelo de clasificación sencillo (árbol de decisión) y se evaluará con un conjunto de test.

2. Algunos nodos de KNIME para preparación de datos

La mayoría de los procesos de preparación de datos se encuentran dentro de las carpetas del repositorio de nodos **Manipulation** y **Analytics**.

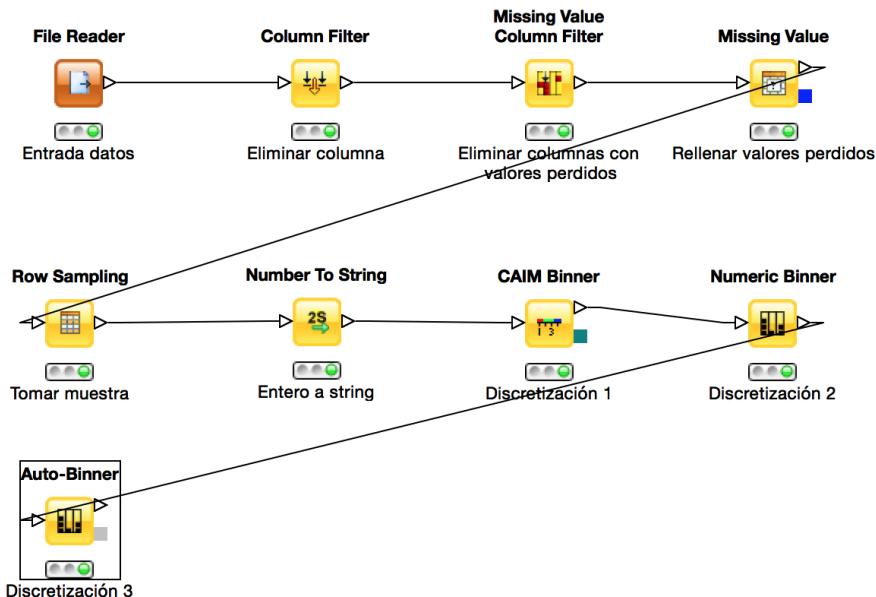


Veamos un ejemplo de su uso. Considera el archivo **autos.txt** que contiene información sobre características de 205 instancias de coches para su uso por una agencia de seguros. La información sobre el archivo y los atributos de las instancias se encuentra en el archivo **autos_names.txt**. Realiza las siguientes operaciones sobre el archivo.

- Carga el archivo utilizando el nodo **IO -> Read -> File Reader**, el lector de archivos por defecto determina que , es la separación entre las columnas.
- Elimina el atributo **normalized-losses** con **Column Filter** en la carpeta **Manipulation/Column/Filter**, ya que contiene muchos valores perdidos
- También podemos eliminar las columnas con algún porcentaje de valores perdidos utilizando **Missing Value Column Filter**, en la misma carpeta. Por ejemplo, si quisieramos eliminar todas las columnas con un 10 % de valores perdidos, bastaría con llenar **Missing value threshold** con 10.
- Otra solución al problema podría ser utilizar el nodo **Missing Value** en **Data Manipulation/Column/Transform**. Es un nodo más completo que los anteriores y permite tanto eliminar los valores perdidos como rellenarlos con ciertos valores predeterminados. Por ejemplo, podemos dar por defecto que los valores numéricos los rellene a la media, y los enteros y string al valor más frecuente. Aunque es posible tratar cada atributo de forma independiente en la opción **Column Settings**.

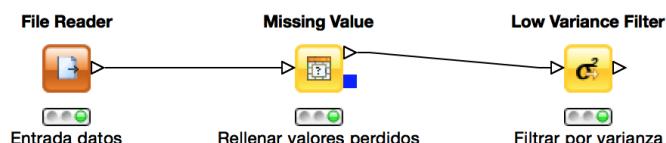
- Realiza un muestreo aleatorio utilizando el nodo **Row Sampling** en **Data Manipulation/Row/Transform**. Por ejemplo, tomando el 30% de las instancias. Para eso basta poner 30 en la casilla **Relative[%]** y seleccionar **Draw randomly**.
- Transforma la primera primera columna (**symboling**) de un tipo de dato entero a cadena de caracteres con **Number to String** en **Manipulation/PMML**. Por ejemplo, esto se podría utilizar si la primera columna es la clase sobre la que queremos discretizar un atributo numérico.
- Discretiza el atributo numérico **wheel-base** utilizando el algoritmo CAIM que ejecuta el nodo **CAIM Binner** en **Manipulation/Column/Binning**. Para el campo **Class column** utiliza **symboling**.
- Discretiza el atributo numérico **width** utilizando el nodo **Numeric Binner** en **Manipulation/Column/Binning**. Analiza el tamaño de los intervalos para que los valores se distribuyan de forma homogénea en tres intervalos.
- Discretiza el atributo numérico **length** utilizando el nodo **Auto Binner** en **Manipulation/Column/Binning**. Selecciona alguna de las configuraciones posibles.

El aspecto del workflow debería terminar de esta manera



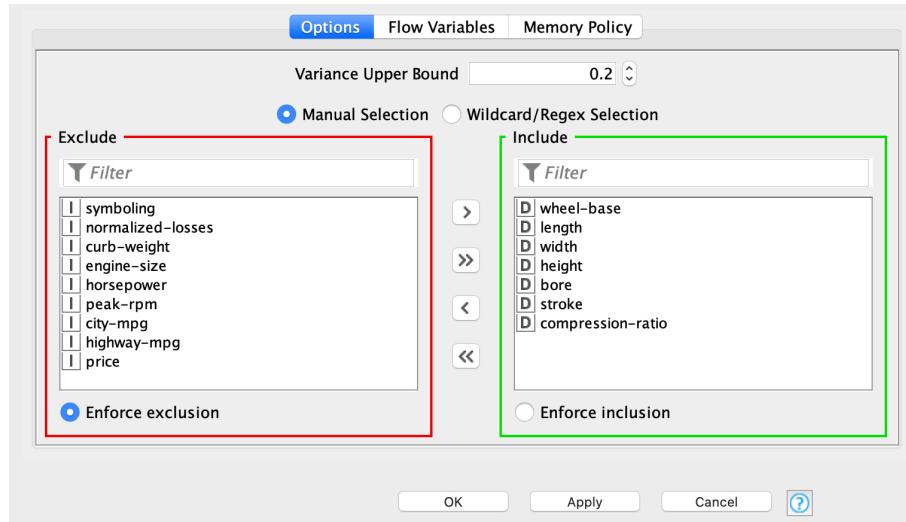
Algo más complicado es utilizar las técnicas de Data Mining para eliminar características o reducir la dimensionalidad. Por ejemplo, podemos utilizar los nodos:

- **Low Variance Filter** en **Analytics/Mining/Feature Selection**. Elimina los atributos numéricos cuya varianza es menor a un cierto umbral. Para realizar un ejemplo, construye un diagrama con la siguiente forma:



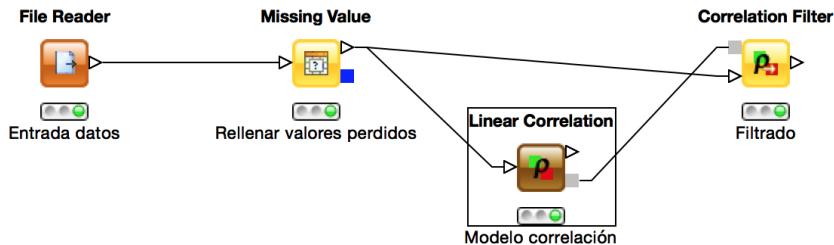
Toma como origen de datos el fichero **autos.txt** y rellena los valores perdidos como se ha hecho anteriormente. En el nodo **Low Variance Filter** selecciona los atributos numéricicos (tipo double) y selecciona

como umbral 0.2.



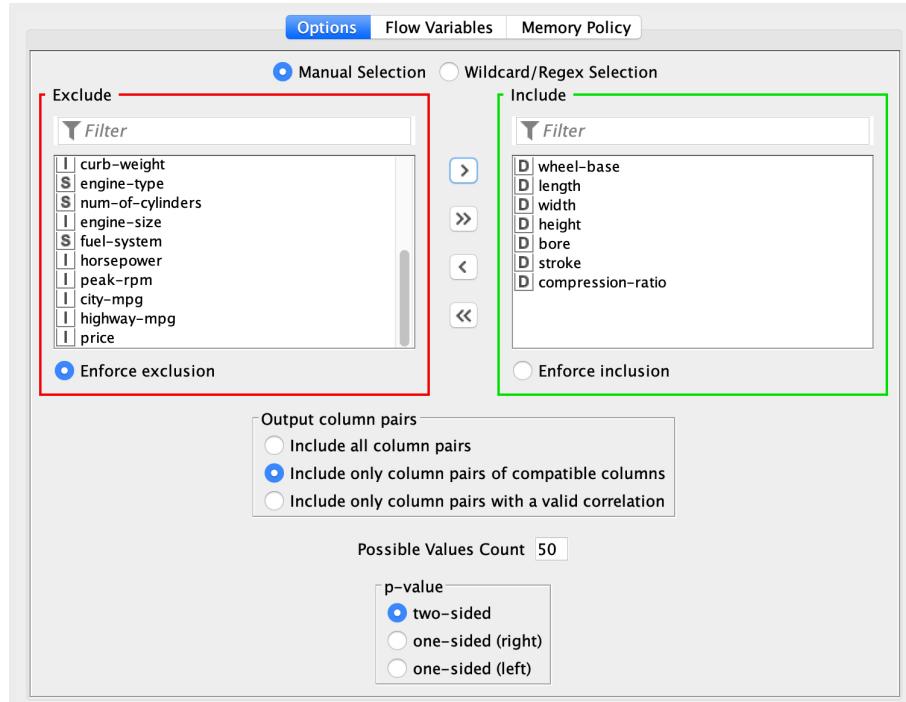
Se deberían haber filtrado **bore** y **stroke**.

- Correlation Filter en Analytics/Mining/Feature Selection y Linear Correlation en Analytics/Statistics. Filtran las columnas que presentan una correlación superior a cierto umbral. Para utilizarlo, construye un diagrama con la siguiente forma:



El nodo **Linear Correlation** calcula las correlaciones y establece el modelo, mientras que el nodo **Correlation Filter** filtra las columnas a partir de los datos y el modelo obtenido (enlace entre los puertos de color gris). En este caso podemos seleccionar los atributos de tipo double (aunque también es

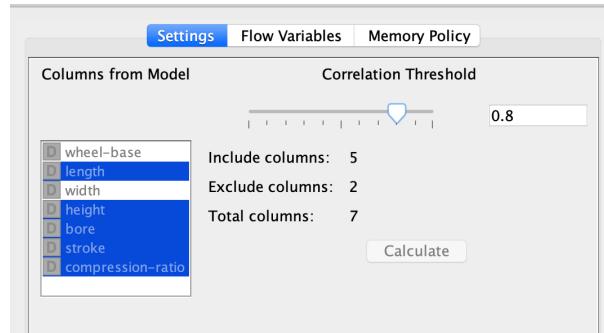
possible utilizar atributos nominales)



obteniendo la siguiente matriz de correlaciones

Row ID	wheel-base	length	width	height	bore	stroke	compression-ratio
wheel-base	1.0	0.87458...	0.7951436...	0.5894347...	0.488760...	0.1609438...	0.249785845...
length	0.874587...	1.0	0.8411182...	0.4910294...	0.606461...	0.1295217...	0.158413706...
width	0.795143...	0.84111...	1.0	0.2792103...	0.559151...	0.1829391...	0.181128626...
height	0.589434...	0.49102...	0.2792103...	1.0	0.171101...	-0.0553513...	0.261214226...
bore	0.488760...	0.60646...	0.5591516...	0.1711013...	1.0	-0.0559089...	0.005200705...
stroke	0.160943...	0.12952...	0.1829391...	-0.055351...	-0.05590...	1.0	0.186105170...
compression-ratio	0.249785...	0.15841...	0.1811286...	0.2612142...	0.005200...	0.1861051...	1.0

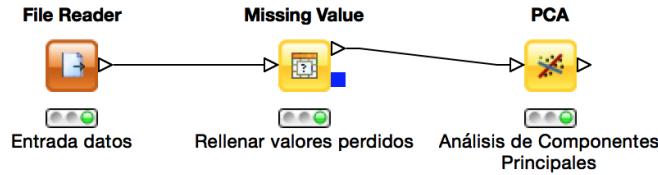
Vemos que la correlación entre las columnas `wheel-base`, `length` y `width` está cercana a uno. En el nodo `Correlation Filter` vamos a seleccionar un umbral de 0.8 en `Correlation Threshold`.



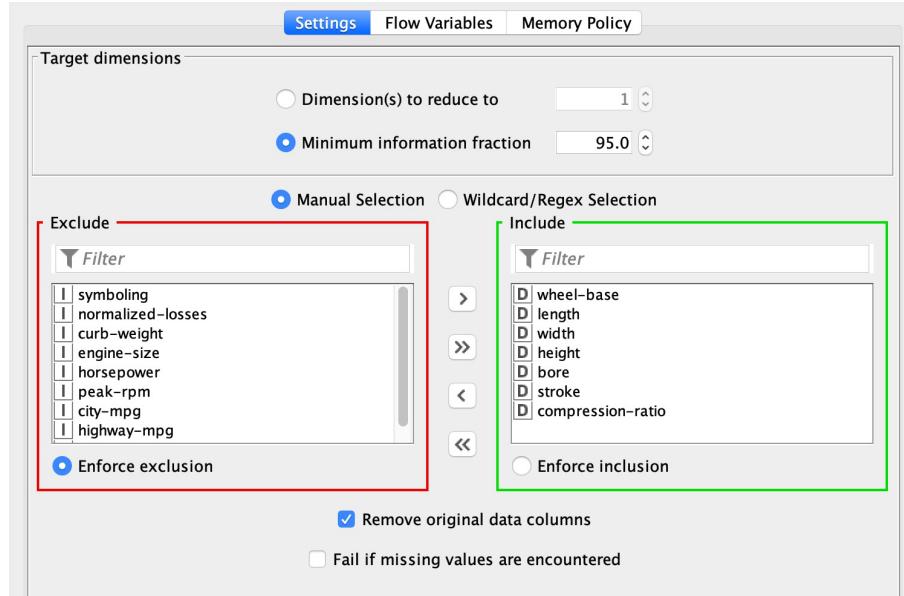
Lo que debería filtrar `wheel-base` y `width`.

- PCA en `Analytics/Mining/PCA`. Realiza un análisis de componentes principales sobre datos numéricos (el resto los deja inalterados). En el fichero `autos.txt`, después de rellenar los valores perdidos con el nodo `Missing Value`, podemos realizar un análisis de componentes principales sobre las columnas de tipo de

dato double. Crea el workflow siguiente:



y configura el nodo PCA para una perdida de información de sólo el 5 %.

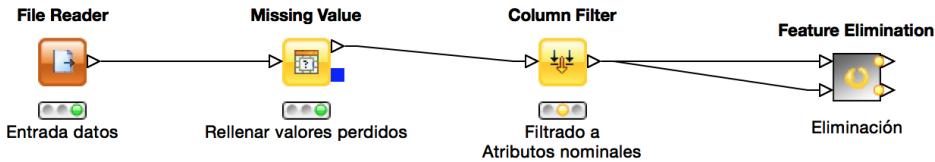


Para conservar las columnas originales no debemos marcar Remove original data columns. En estas condiciones reducimos los siete atributos a sólo tres nuevos atributos construidos a partir de los anteriores.

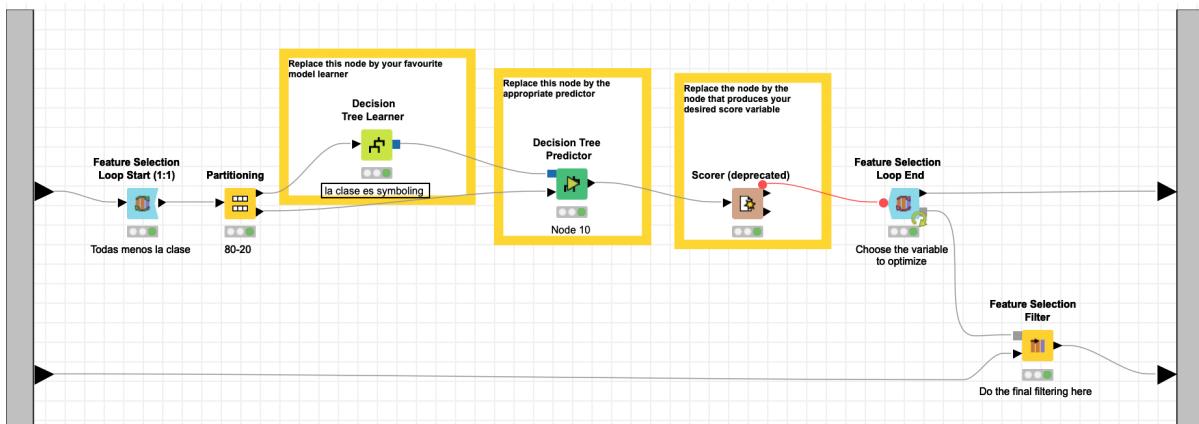
price	PCA d...	PCA d...	PCA d...
195	-9.594	-3.211	-7.501
500	-9.594	-3.211	-7.501
500	-4.532	-1.723	-2.407
950	2.796	-0.258	0.141
150	2.546	-2.236	0.461
250	3.238	-1.998	-0.113
710	20.42	-2.968	-0.232
320	20.42	-2.968	-0.232
375	20.427	-3.124	-0.089
500	3.948	-3.819	-0.684
130	3.288	-1.131	1.587
325	3.288	-1.131	1.587
370	3.298	-0.936	1.523
105	3.298	-0.936	1.523
665	15.596	-2.357	-0.677
760	15.543	-3.302	-0.363
315	19.797	-4.482	-2.945
380	25.946	-3.184	1.87
51	-34.615	2.858	4.209
95	-18.524	1.479	2.915

- Backward Feature Elimination en Analytics/Mining/Feature Selection/Meta nodes. Un metanodo no es más que un nodo que contiene un flujo modificado por un conjunto de nodos. Para acceder a los

nodos internos al metanodo sólo debemos realizar doble click sobre el metanodo. Por ejemplo, consideremos el siguiente flujo para eliminación de características (Feature Elimination)

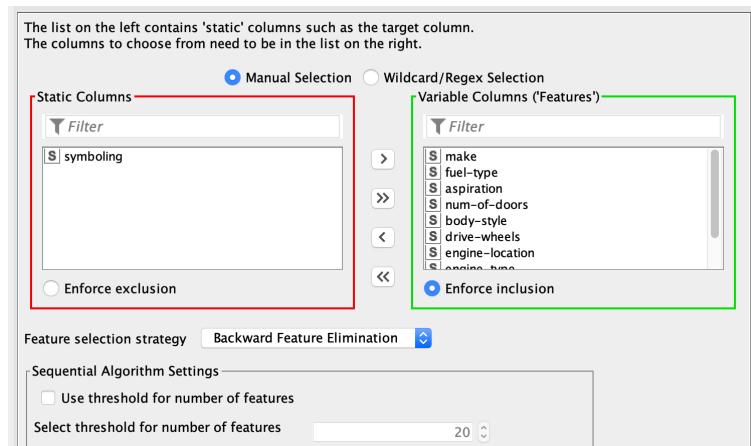


donde primeramente hemos filtrado los atributos nominales (el atributo `symboling` hace de clase a la que clasificar, que se ha transformado a nominal). El metanodo **Backward Feature Elimination** reduce la dimensionalidad basándose en un algoritmo de clasificación (por defecto utiliza Naïve-Bayes). La composición interna del metanodo es la siguiente



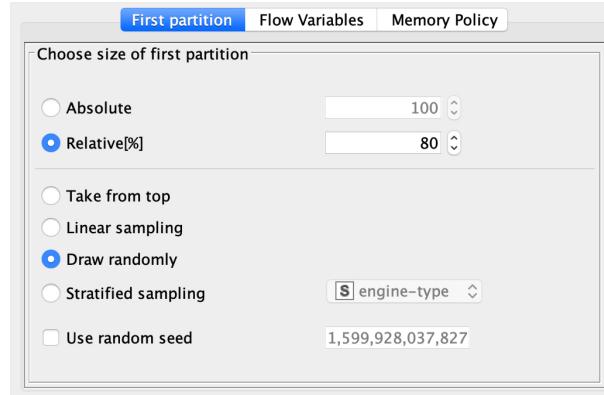
donde hemos cambiado el clasificador Naïve-Bayes por un clasificador basado en árboles de decisión. La configuración de los nodos es la siguiente:

- **Feature Selection Loop Start.** Seleccionamos todos los atributos nominales (los que no son nominales se han filtrado fuera del metanodo), menos la clase sobre la que clasificamos.

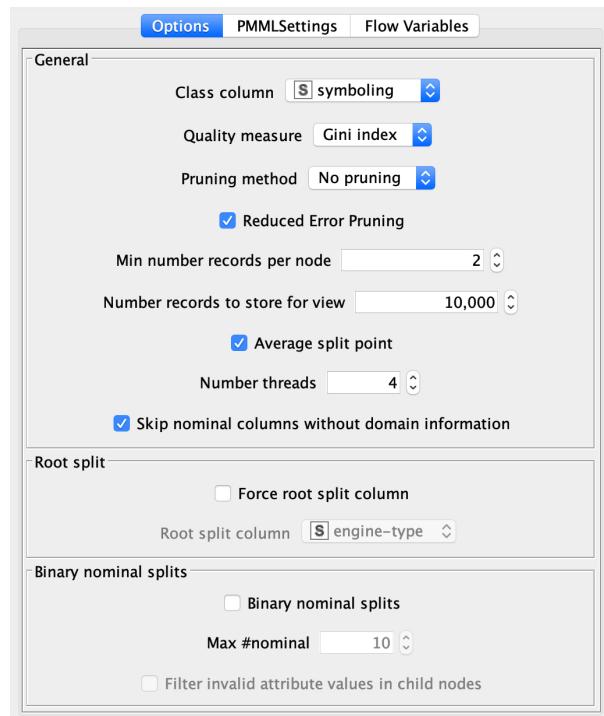


- **Partitioning.** Una partición estándar para un conjunto de entrenamiento con el 80 % de las instan-

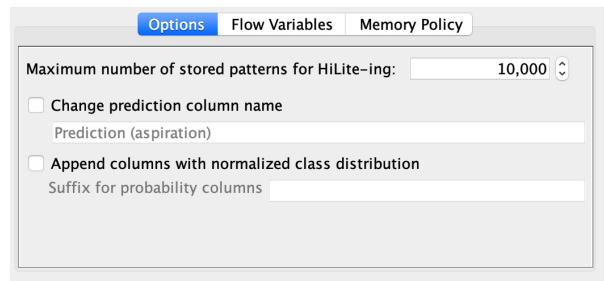
cias.



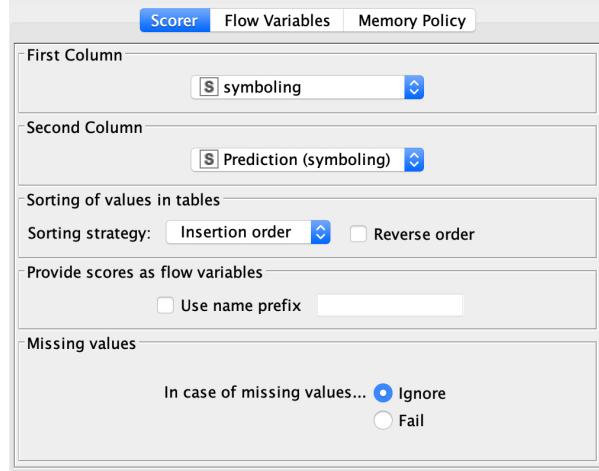
- Decision Tree Learned



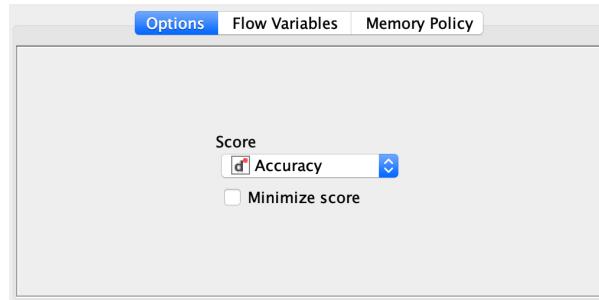
- Decision Tree Predictor



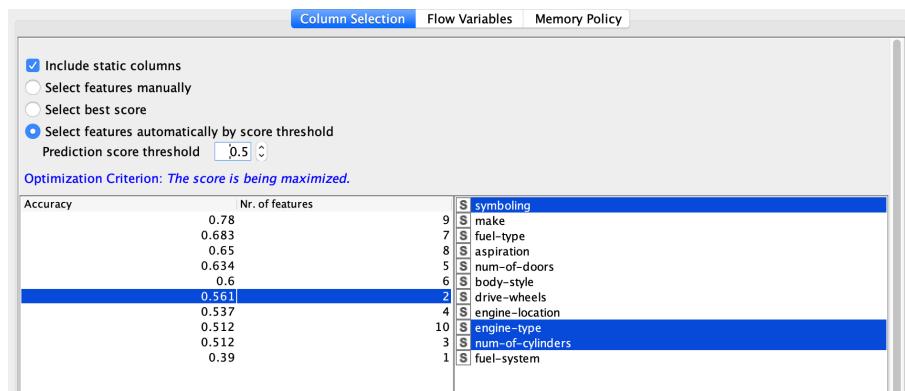
- Scorer



- Feature Selection Loop End. El funcionamiento del nodo es sencillo, clasifica las instancias usando N atributos respecto a un atributo clase. Después ejecuta $N - 1$ veces la misma operación sin considerar cada vez uno de los atributos. El atributo con menor error respecto a la clasificación considerando todos los atributos es eliminado. Este proceso se realiza hasta que sólo queda un atributo con el que conseguir la clasificación. Este proceso se realiza hasta que sólo queda un atributo con el que conseguir la clasificación.



- Feature Selection Filter, para filtrar las columnas dependiendo del error, de forma manual o mediante un error umbral.



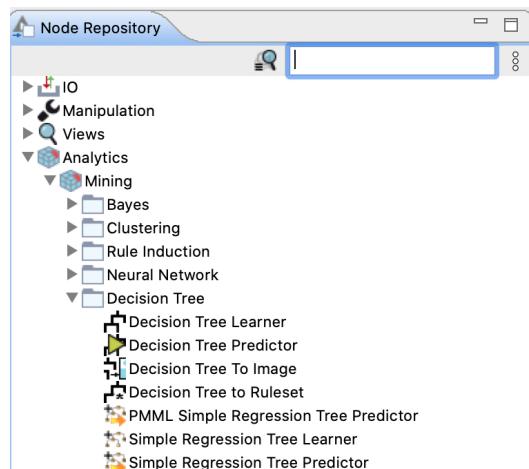
Por ejemplo, para un umbral de 0.5, nos quedaríamos con las columnas `symboling`, `engine-type` y `num-of-cylinders`.

3. Descripción de la base de datos: gravedad en accidentes de tráfico

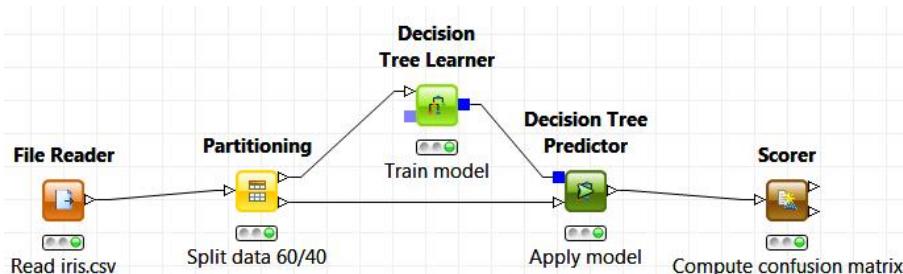
En EE.UU., el General Estimate System (GES) es un componente del National Automotive Sampling System (NASS) mantenido por la Administración Nacional de Seguridad del Tráfico en Carreteras. El GES obtiene sus datos de una muestra representativa a nivel nacional de los aproximadamente 6,4 millones de accidentes informados por la policía que se producen anualmente. Estos accidentes incluyen aquellos que resultan mortales o causan lesiones y los relacionados con daños materiales. Al restringir la atención a los accidentes informados por la policía, el GES se concentra en los accidentes de mayor preocupación para la comunidad de seguridad vial y el público en general. GES se utiliza para identificar áreas con problemas de seguridad vial, proporcionar una base para iniciativas de información al consumidor y normativas, así como facilitar el análisis de costes y beneficios de las iniciativas de seguridad en carretera. En esta práctica, utilizaremos como ejemplo el conjunto de datos tomado como muestra en el año 2001 (55.964 datos) y disponible al público en http://www.transtats.bts.gov/Fields.asp?Table_ID=1158. Para facilitar las pruebas durante la realización de la práctica, dada la cantidad de datos, se facilitará el archivo completo .xls, y una versión con un número reducido de instancias. Ambos archivos .xls contienen dos hojas: una, con la descripción de las variables y los posibles valores que pueden tomar, y, otra, con los datos recogidos.

El objetivo del problema es el de predecir la gravedad del daño causado en un accidente (mortal, lesiones o daños materiales) en función de una serie de características como la ingesta de alcohol del conductor, hora del accidente, condición de la carretera, etc. Este tipo de predicción podría ser útil, por ejemplo, para priorizar la dotación de recursos en respuesta a un accidente. Sin embargo, los datos disponibles presentan importantes deficiencias (valores perdidos, características con excesivas categorías, características e instancias prescindibles, etc.) que sugieren la necesidad de ser preprocesados antes de aplicar otras técnicas de Minería de Datos.

La práctica consiste en preparar los datos disponibles para su posterior análisis mediante técnicas de clasificación. Utilizaremos el algoritmo C4.5 como técnica de clasificación para probar los beneficios obtenidos en cada mejora aplicada sobre los datos. El algoritmo de clasificación se encuentra implementado en el nodo de Knime “Decision Tree Learner” en Analytics/Mining/Decision Tree.



Por otro lado, dada la dimensión del problema y los numerosos experimentos a realizar, es mejor considerar una única partición de datos de entrenamiento y prueba (hold-out) en toda la experimentación (por ejemplo, 80 %-20 % o 60 %-40 %). El ejemplo más básico de flujo para construir y evaluar un árbol de decisión es del tipo



Antes de iniciar cualquier otra tarea:

1. deberán dejarse en blanco las celdas que contengan valores desconocidos para su correcto tratamiento posterior. En la descripción de cada variable incluida en el archivo .xls se describe cómo se indica que se trata de un valor desconocido. Normalmente se emplean valores tales como “9” o “99”. En aquellas características que contienen valores perdidos se incluyen dos variables (es decir, dos columnas). Una de ellas contiene los valores en bruto y, por tanto, conserva los valores desconocidos. La otra variable, cuyo nombre siempre es igual al original (o su abreviatura) más la cadena _I al final (ojo!, la velocidad máxima permitida es con _H), contiene valores imputados, es decir, los valores desconocidos se han sustituidos por un valor válido según algún criterio. Es importante señalar que nunca deberán usarse una variable y su imputada simultáneamente en ningún experimento.
2. Hay que construir la variable clase (que es la que se clasificará) como combinación de las tres variables que describen la gravedad del accidente (FATALITIES, INJURY CRASH y PRPTYDMG CRASH). Se deja como elección del alumno cómo construir esa variable, pero debería depender de las tres anteriores.

Ambos tratamientos se pueden realizar directamente con un editor de hojas de cálculo (por ejemplo, *Microsoft Excel*).

Algunos procesos posibles a realizar para el tratamiento de los datos:

1. **Discretización.** Algunas pocas características (por ejemplo, hora del accidente o velocidad máxima permitida) contienen valores discretos pero según una escala ordenada y de suficiente cardinalidad como para plantearse la conveniencia de discretizarlos o reducirlos a un número inferior de valores posibles. Además, varias características tienen un elevado número de categorías posibles que dificultan el proceso de aprendizaje del clasificador. Se deberá trabajar con las variables con valores imputados cuyos nombres acaban en _I.
 - a) Ejecutar el algoritmo de prueba para clasificación (C4.5) y estudiar cómo divide las características numéricas en los árboles de decisión aprendidos.
 - b) Aplicar el algoritmo de discretización top-down CAIM sobre las características numéricas y comprobar el comportamiento del algoritmo de prueba.
 - c) Estudiar las distintas características categóricas y proponer una discretización de las mismas basándose en el significado de la característica, la visualización de los datos, etc. Del mismo modo, proponer una discretización de las características numéricas basándose en su significado (por ejemplo, accidentes en hora punta o no). Estudiar los resultados sobre el algoritmo de prueba.
2. **Valores perdidos.** El conjunto de datos contiene valores perdidos, es decir, valores desconocidos en algunas instancias para algunas características. Es conveniente trabajar con el conjunto de datos con características discretizadas obtenido en 1 para reducir el esfuerzo computacional.
 - a) Ejecutar el algoritmo de prueba para clasificación (C4.5) sobre los datos sin imputar y comprobar el comportamiento de este algoritmo para tratar implícitamente datos perdidos.
 - b) Ejecutar el algoritmo de prueba con los datos imputados disponibles y comprobar su comportamiento.
 - c) Imputar valores perdidos con la media o moda, según proceda, y comprobar el comportamiento del algoritmo de prueba.
 - d) Eliminar las instancias que contienen algún valor perdido y comprobar el comportamiento del algoritmo de prueba.
 - e) Eliminar las características con valores perdidos y comprobar el comportamiento del algoritmo de prueba.
 - f) Emplear un algoritmo de predicción (clasificación o regresión, según la naturaleza de la variable) para imputar valores perdidos y comprobar su comportamiento con el algoritmo de prueba.
3. **Selección de características.** El conjunto de datos contiene un gran número de características, algunas de las cuales podrían prescindirse en aras de facilitar la interpretabilidad del conocimiento extraído e incluso mejorar la precisión. Se puede trabajar con el conjunto de datos con características discretizadas obtenido en 1 para reducir el esfuerzo computacional.

- a) Ejecutar el algoritmo de prueba para clasificación (C4.5) sobre el conjunto de datos completo. El modelo generado por este algoritmo es posible que no emplee todas las características disponibles, por lo que ya estará realizando una selección de ellas de forma implícita.
 - b) Aplicar una selección de características envolvente hacia atrás (backward). También se puede decidir eliminar o forzar a conservar, según el caso, algunas características basándose en algún criterio (por ejemplo, tras una visualización de datos) con el objetivo de reducir el coste computacional.
4. **Selección de instancias.** El conjunto de datos contiene un número considerable de instancias. Reducir este número no solo ayuda a mejorar el coste computacional del algoritmo de aprendizaje, sino que también puede ayudar a generar modelos más legibles (por ejemplo, árboles con menos nodos) e incluso más precisos. Una buena idea es trabajar con el conjunto de datos con características seleccionadas obtenido en 3 para reducir el esfuerzo computacional siempre que se haya demostrado que se mejora la precisión en la clasificación.
- a) Aplicar técnicas de muestreo y comprobar su comportamiento en el algoritmo de prueba (C4.5).
 - b) El conjunto de datos contiene una categoría de la clase mucho más infrecuente que el resto, lo que hace que los datos no estén balanceados. Analizar esta situación, realizar una reducción de datos mediante muestreo aleatorio que equilibre la frecuencia de la clase y comprobar el efecto en el algoritmo de prueba. Para este análisis se puede reducir el problema a una clase binaria (por ejemplo, clase minoritaria frente al resto) y estudiar la matriz de confusión obtenida para los modelos aprendidos con y sin datos balanceados.