Attribute Los filtros agrupados en esta categoría son aplicados a atributos.

Add Añade un atributo más. Como parámetros debemos proporcionarle la posición que va a ocupar este nuevo atributo (esta vez comenzando desde el 1), el nombre del atributo y los posibles valores de ese atributo separados entre comas. Si no se especifican, se sobreentiende que el atributo es numérico.

AddExpression Este filtro es muy útil puesto que permite agregar al final un atributo que sea el valor de una función. Es necesario especificarle la fórmula que describe este atributo, en donde podemos calcular dicho atributo a partir de los valores de otro u otros, refiriéndonos a los otros atributos por "a" seguido del número del atributo (comenzando por 1).

Por ejemplo:

```
(a3^3.4)*a1+sqrt(floor(tan(a4)))
```

Los operadores y funciones que soporta son +, −, *, /, ^, *log*, *abs*, *cos*, *exp*, *sqrt*, *floor* (función techo), *ceil* (función suelo), *rint* (redondeo a entero), *tan*, *sin*, (,).

Otro argumento de este filtro es el nombre del nuevo atributo.

AddNoise Añade ruido a un determinado atributo que debe ser nominal. Podemos especificar el porcentaje de ruido, la semilla para generarlo y si queremos que al introducir el ruido cuente o no con los atributos que faltan.

ClusterMembership Filtro que dado un conjunto de atributos y el atributo que define la clase de los mismos, devuelve la probabilidad de cada uno de los atributos de estar clasificados en una clase u otra.

Tiene por parámetro *ignoredAttributeIndices* que es el rango de atributos que deseamos excluir para aplicar este filtro. Dicho intervalo podemos expresarlo por cada uno de los índices los atributos separados por comas o definiendo rangos con el símbolo guión ("-").

Es posible denotar al primer y último atributo con los identificadores *first* y *last* (en este caso la numeración de los atributos comienza en 1, por lo que *first* corresponde al atributo número 1).

Copy Realiza una copia de un conjunto de atributos en los datos. Este filtro es útil en conjunción con otros, ya que hay ciertos filtros (la mayoría) que destruyen los datos originales. Como argumentos toma un conjunto de atributos expresados de la misma forma que el filtro anterior. También tiene una opción que es *invertSelection* que invierte la selección realizada (útil para copiar, por ejemplo, todos los atributos menos uno).

Discretize Discretiza un conjunto de valores numéricos en rangos de datos. Como parámetros toma los índices de los atributos discretizar (*attribute indices*) y el número de particiones en que queremos que divida los datos (*bins*). Si queremos que las particiones las realice

por la frecuencia de los datos y no por el tamaño de estas tenemos la opción *useEqualFrecuency*. Si tenemos activa esta última opción podemos variar el peso de las instancias para la definición de los intervalos con la opción *DesiredWeightOfInstancesPerInterval*. Si, al contrario tenemos en cuenta el número de instancias para la creación de intervalos podemos usar *findNumBins* que optimiza el procedimiento de confección de los mismos.

Otras opciones son *makeBinary* que convierte los atributos en binario e *invertSelection* que invierte el rango de los atributos elegidos.

FistOrder Este filtro realiza una transformación de los datos obteniendo la diferencia de pares consecutivos de datos, suponiendo un dato inicial adicional de valor 0 para conseguir que la cardinalidad del grupo de datos resultante sea la misma que la de los datos origen.

Por ejemplo, si los datos son 1 5 4 6, el resultado al aplicar este filtro será 1 4 -1 2. Este filtro toma un único parámetro que es el conjunto de atributos con el que obtener esta transformación.

MakeIndicator Crea un nuevo conjunto de datos reemplazando un atributo nominal por uno booleano (Asignará "1" si en una instancia se encuentra el atributo nominal seleccionado y "0" en caso contrario).

Como atributos este filtro toma el índice el atributo nominal que actuará como indicador, si se desea que la salida del filtro sea numérica o nominal y los índices los atributos sobre los que queremos aplicar el filtro.

MergeTwoValues Fusiona dos atributos nominales en uno solo. Toma como argumentos la posición del argumento resultado y la de los argumentos fuente.

NominalToBinary Transforma los valores nominales de un atributo en un vector cuyas coordenadas son binarias.

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

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

NumericToBinary Convierte datos en formato numérico a binario. Si el valor de un dato es 0 o desconocido, el valor en binario resultante será el 0.

NumericTransform Filtro similar a `AddExpression` pero mucho más potente. Permite aplicar un método java sobre un conjunto de atributos dándole el nombre de una clase y un método.

Obfuscate Ofusca todas las cadenas de texto de los datos. Este filtro es muy útil si se desea compartir una base de datos pero no se quiere compartir información privada.

PKIDiscretize Discretiza atributos numéricos (al igual que `Discretize`), pero el número de intervalos es igual a la raíz cuadrada del número de valores definidos.

RandomProjection Reduce la dimensionalidad de los datos (útil cuando el conjunto de datos es muy grande) proyectándola en un subespacio de menor dimensionalidad utilizando para ello una matriz aleatoria. A pesar de reducir la dimensionalidad los datos resultantes se

procura conservar la estructura y propiedades fundamentales de los mismos.

El funcionamiento se basa en el siguiente producto de matrices:

$$X(i \times n) * R(n \times m) = Xrp(i \times m)$$

Siendo X la matriz de datos original de dimensiones i (número de instancias) $\times$ n (número de atributos), R la matriz aleatoria de dimensión n (número de atributos) $\times$ m (número de atributos reducidos) y Xrp la matriz resultante siendo de dimensión $i \times m$.

Como parámetros toma el número de parámetros en los que queremos aplicar este filtro (*numberOfAttributes*) y el tipo de distribución de la matriz aleatoria que puede ser:

Sparse 1 $-\sqrt{3}$ con probabilidad $\frac{1}{6}$, 0 con probabilidad $\frac{2}{3}$ y $\sqrt{3}$ con probabilidad $\frac{1}{6}$.

Sparse 3 -1 con probabilidad $\frac{1}{2}$ y 1 con probabilidad $\frac{1}{2}$.

Gaussian Utiliza una distribución gaussiana.

Además de estos parámetros, pueden utilizarse también el número de atributos resultantes después de la transformación expresado en porcentaje del número de atributos totales (*percent*), la semilla usada para la generación de números aleatorios (*seed*), y si queremos que aplique antes de realizar la transformación el filtro `ReplaceMissingValues`, que será explicado más adelante.

Remove Borra un conjunto de atributos del fichero de datos.

RemoveType Elimina el conjunto de atributos de un tipo determinado.

RemoveUseless Elimina atributos que oscilan menos que un nivel de variación. Es útil para eliminar atributos constantes o con un rango muy pequeño. Como parámetro toma el máximo porcentaje de variación permitido, si este valor obtenido es mayor que la variación obtenida la muestra es eliminada.

$$\text{Variación obtenida} = \frac{\text{Número de atributos distintos}}{\text{Número de atributos}} * 100$$

ReplaceMissingValues Reemplaza todos los valores indefinidos por la moda en el caso de que sea un atributo nominal o la media aritmética si es un atributo numérico.

Standarize Estandariza los datos numéricos de la muestra para que tengan de media 0 y la unidad de varianza. Para estandarizar un vector x se aplica la siguiente fórmula:

$$x(i) = \frac{x(i) - \overline{x}}{\sigma(x)}$$

StringToNominal Convierte un atributo de tipo cadena en un tipo nominal.

StringToWordVector Convierte los atributos de tipo `String` en un conjunto de atributos representando la ocurrencia de las palabras del texto. Como atributos toma:

- *DFTranform* que indica si queremos que las frecuencias de las palabras sean transformadas según la regla:

$$\text{frec. de la palabra i en la instancia j} * \log \frac{\text{nº de instancias}}{\text{nº de instancias con la palabra i}}$$

- *TFTransform* Otra regla de transformación:

$$\log(1 + \text{frecuencia de la palabra i en la instancia j})$$

- *attributeNamePrefix*, prefijo para los nombres de atributos creados
- Delimitadores (*delimiters*), conjunto de caracteres escape usados para delimitar la unidad fundamental (*token*). Esta opción se ignora si la opción *onlyAlphabeticTokens* está activada, ya que ésta, por defecto, asigna los *tokens* a secuencias alfabéticas usando como delimitadores todos los caracteres distintos a éstos.
- *lowerCaseTokens* convierte a minúsculas todos los tokens antes de ser añadidos al diccionario.
- *normalizeDocLength* selecciona si las frecuencias de las palabras en una instancia deben ser normalizadas o no.
- *outputWordCounts* cuenta las ocurrencias de cada palabra en vez de mostrar únicamente si están o no están en el diccionario.
- *useStoplist* si está activado ignora todas las palabras de una lista de palabras excluidas (*stoplist*).
- *wordsToKeep* determina el número de palabras (por clase si hay un asignador de clases) que se intentarán guardar.

SwapValues Intercambia los valores de dos atributos nominales.

TimeSeriesDelta Filtro que asume que las instancias forman parte de una serie temporal y reemplaza los valores de los atributos de forma que cada valor de una instancia es reemplazado con la diferencia entre el valor actual y el valor pronosticado para dicha instancia.

En los casos en los que la variación de tiempo no se conozca puede ser que la instancia sea eliminada o completada con elementos desconocidos (símbolo "?"). Opciones:

- *attributeIndices* Especifica el rango de atributos en los que aplicar el filtro.
- *FillWithMissing* Las instancias al principio o final del conjunto de datos, donde los valores calculados son desconocidos, se completan con elementos desconocidos (símbolo "?") en vez de eliminar dichas instancias, que es el comportamiento por defecto.
- *InstanceRange* Define el tamaño del rango de valores que se usará para realizar las restas. Si se usa un valor negativo significa que realizarán los cálculos con valores anteriores.
- *invertSelection* Invierte la selección realizada.

Instance Los filtros son aplicados a instancias concretas enteras.

NonSparseToSparse Convierte una muestra de modo completo a modo abreviado.

Randomize Modifica el orden de las instancias de forma aleatoria.

RemoveFolds Permite eliminar un conjunto de datos. Este filtro está pensado para eliminar una partición en una validación cruzada.

RemoveMisclassified Dado un método de clasificación lo aplica sobre la muestra y elimina aquellas instancias mal clasificadas.

RemovePercentage Suprime un porcentaje de muestras.

RemoveRange Elimina un rango de instancias.

RemoveWithValues Elimina las instancias acordes a una determinada restricción.

Resample Obtiene un subconjunto del conjunto inicial de forma aleatoria.

SparseToNonSparse Convierte una muestra de modo abreviado a modo completo.Es la operación complementaria a `NonSparseToSparse`.