Organización de variables

1) Reordenar:
Filter → unsupervised → attribute → Reorder
Esta operación es especialmente útil si nuestra variable respuesta no se encuentra en la última posición.
En attributeIndices especificamos el nuevo orden de las variables. Por ejemplo si nuestra variable respuesta se encuentra en segunda posición, y queremos colocarla al final, lo indicaríamos así: 1,3-last,2

Técnicas de tratamiento de datos perdidos

1) Eliminación de instancias con datos perdidos:
Filter → Unsupervised → Instance → RemoveWithValues.
En el filtro, definir el parámetro attributeIndicies como el índice del atributo con valores perdidos que se va a usar para limpiar y definir matchMissingValues como True. Si se quieren eliminar de todos los atributos, repetir esto cambiando el valor de attributeIndicies.

2) Imputación por media o moda (recomendado):
Filter → Unsupervised → Attribute → ReplaceMissingValues.

Tratamiento de atributos categóricos para transformarlos en atributos binarios

1) One Hot Encoding.
Filter → Unsupervised → Attribute → NominaltoBinary
Definir como índice el del atributo a binarizar, tener definida la clase de salida a la que se quiera predecir y establecer binaryAtributtesNominal a True.

Tratamiento adicional para modelos no basados en árboles

1) Normalización:
Filter → Unsupervised → Attribute →Normalize.
En scale se indica la amplitud del rango desde el extremo inferior al superior y en translation, el extremo inferior.

2) Estandarización:
Filter→ Unsupervised → Attribute → Standarize.

3) Discretización de variables numéricas en rangos categóricos:
Filter → Unsupervised → Attribute →Discretize.
Ajustar el número de bins: número de particiones que se va a hacer sobre el rango completo.
Especificar los índices de los atributos a discretizar.

Mediante la propiedad useEqualFrequency, indicar si las particiones deben tener la misma amplitud (false) o el mismo numero de instancias (true).

Clasificadores típicos para emplear

ZeroR como clasificador baseline antes de probar el resto. Random Forest, J48, NaiveBayes, IBK, JRIP, MultilayerPerceptron, regresión logística.

Recordemos que es recomendable evaluar los modelos mediante validación cruzada usando un número de folds entre 5 y 10.

Selección de atributos

Pestaña Select attributes, se puede dejar el CfsSubsetEval (no admite cadenas de texto) y el método de búsqueda por defecto. Seleccionar la clase a predecir y darle a start. Si se evalúa sobre el conjunto de entrenamiento completo devuelve los atributos a mantener, si se usa validación cruzada, devuelve el porcentaje de los folds en los que se ha elegido ese atributo (mayor porcentaje, mayor confianza).

Clustering

Pestaña Cluster, elegir SimpleKMeans, definir el número de clusters que se quieren crear y el tipo de distancia. Se pueden ignorar atributos durante el cálculo de los clusters seleccionándolos en Ignore attributes. Los resultados indican la media o moda de los atributos en cada grupo final, lo que sirve para caracterizarlos y después se puede interpretar representando los atributos que generen las diferencian con botón derecho sobre los resultados → Visualize cluster assignments y seleccionando los atributos de interés en los ejes.

Optimización de hiperparámetros

En los clasificadores, Meta → CVParameterSelection. Definir el clasificador a optimizar en las opciones, abrir el ArrayEditor y escribir el nombre del hiperparámetro a optimizar, su valor inferior, su valor superior, y el número de valores que coger en ese rango y pulsar Add. Ej: C 0.1 0.5 5 en un J48 probará los valores 0.1, 0.2, 0.3, 0.4 y 0.5 para el parámetro de confianza (C) de ese clasificador. La salida devuelve los óptimos. Nótese que Weka solo permite optimizar hiperparámetros numéricos.