TEMA 4. PREPROCESADO DE DATOS

Contenidos

I. Introducción
II. Valores ausentes
III. Detección de Outliers
IV. Creación de nuevos atributos
V. Normalización y estandarización
VI. Discretización de atributos continuos
VII. “Continuización” de atributos discretos
VIII. Preprocesado en Orange
IX. Actividades

I. Introducción

¿Por qué preprocesar los datos?

Los datos de los problemas reales suelen ser:

- **Datos incompletos:** valores de atributos inexistentes (ausentes/missings)
- **Datos con ruido:** errores de precisión, errores de medición, errores de almacenamiento
- **Datos inconsistentes:** outliers
- **Datos con formato no compatible**
- **…**

Para obtener conclusiones válidas y útiles es necesario una adecuada preparación previa de los datos.

II. Valores ausentes

La presencia de datos faltantes o perdidos puede ser un problema que puede conducir a resultados poco precisos:

- Algunos valores ausentes expresan características relevantes
- Datos incompletos

Posibles soluciones:

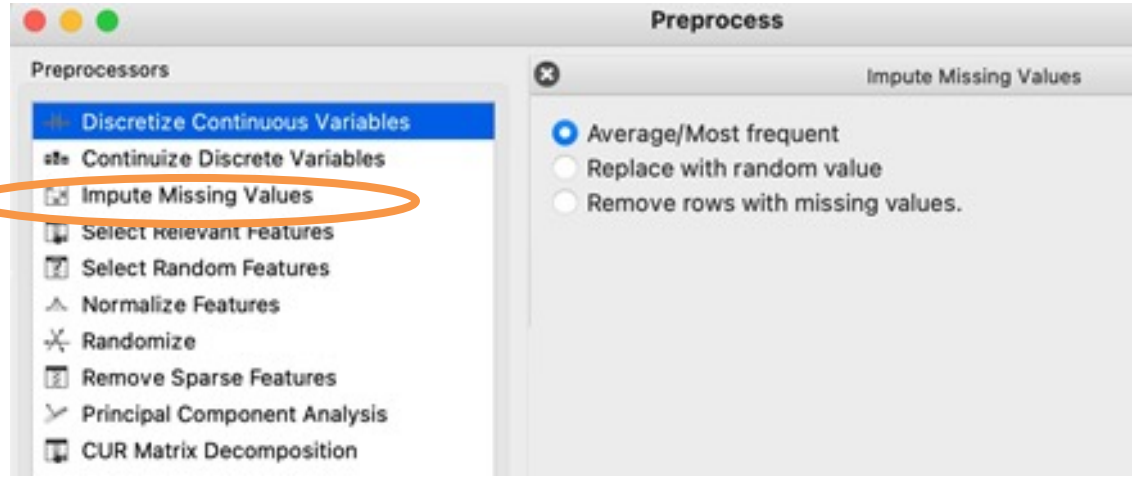
- Eliminar ejemplos con atributos sin valor: Pérdida de información
- Asignar una constante o un valor aleatorio: Mala interpretación del algoritmo de aprendizaje
- Reemplazar el valor: por medias o modas (demasiado *naive*) o predecir el valor (puede ser costoso)

II. Valores ausentes

Nodo Preprocess

Impute Missing Values

- “Average/Most frequent”*:* reemplaza los valores ausentes (**?**) por la media (para valores numéricos) o por la moda (valores cualitativos).
- “Replace with random value”*:* reemplaza los valores ausentes con valores aleatorios, siempre entre el rango de la variable en cuestión.
- “Remove rows with missing values”: elimina las filas con algún valores ausentes.

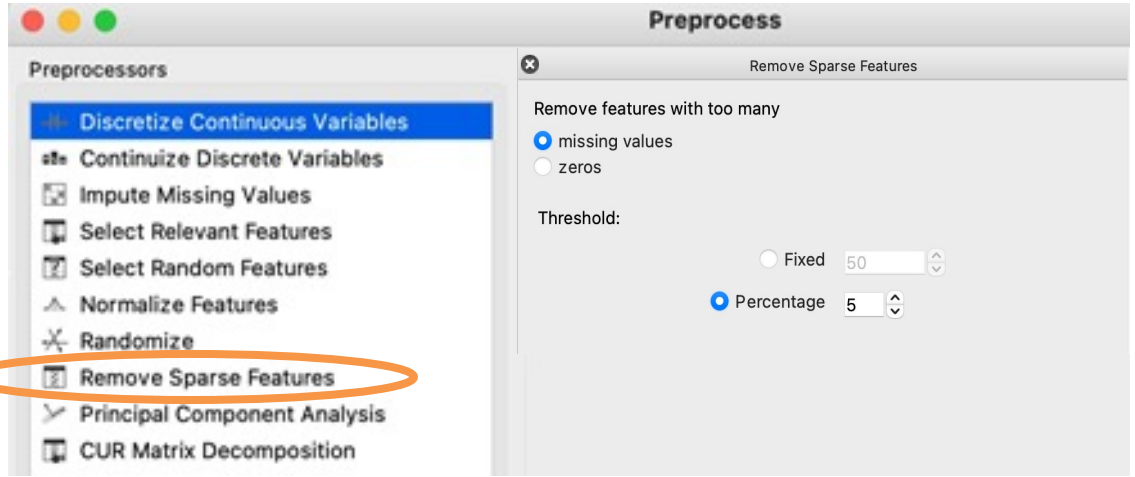


II. Valores ausentes

Nodo Preprocess

Remove Sparse Features

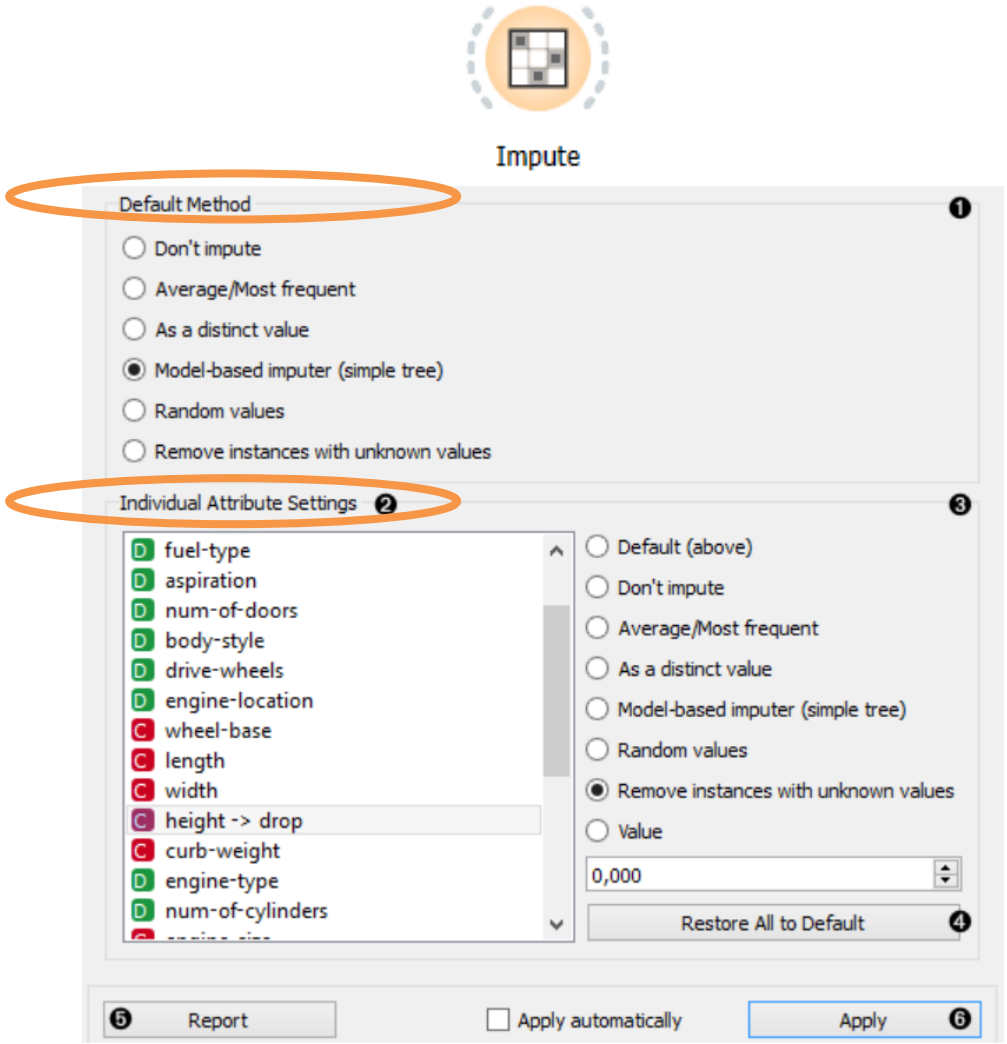
- Conserva solo aquellos **atributos** que superan un umbral de porcentaje o valor fijo de valores no nulos (o ceros).
- Hay que tener cuidado al usarlo.



II. Valores ausentes

Nodo Impute

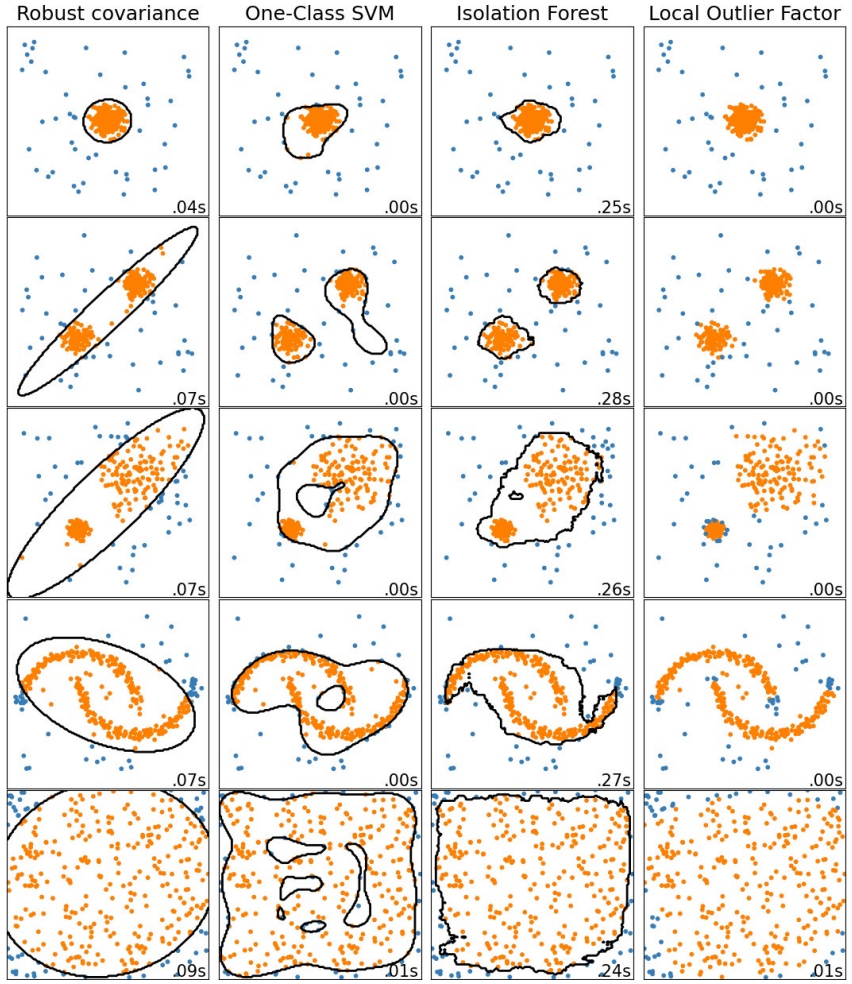
- "Model-based imputer": Construye un modelo para predecir los valores ausentes basándose en los valores de otros atributos.
  - El modelo por defecto es 1-NN (1-nearest neighbor): toma el valor de la instancia más similar.
- Cada atributo se puede imputar por separado.



III. Detección de outliers

Para la detección de outliers se suelen aplicar técnicas de aprendizaje no supervisado:

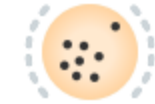
- Estimador de covarianza: Ajusta una elipse a los puntos centrales con la distancia Mahalanobis (solo datos con distribución Gaussiana)
- One-class SVM with non-linear kernel (RBF): Clasifica los datos como similares o diferentes. Suele funcionar bien para distribuciones no Gaussianas.
- Isolation Forest: Usa Random Forest: aísla aleatoriamente un atributo y luego selecciona aleatoriamente un valor de división (entre su máximo y mínimo).
- Local Outlier Factor: Usa K-Nearest Neighbors. Algoritmo que mide el grado de anormalidad mediante la desviación de un punto con respecto a sus vecinos.



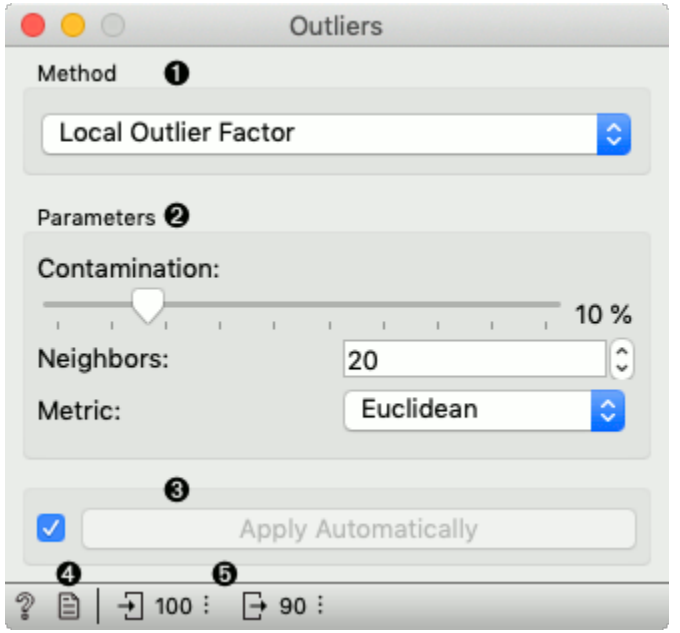
III. Detección de outliers

Cada método tiene unos parámetros específicos. Por ejemplo el método Local Outlier Factor:

- "Contamination": Proporción de outliers en los datos.
- "Neighbors": Número de vecinos a tener en cuenta.
- "Metric": Métrica para calcular la distancia entre los vecinos.



Outliers



IV. Creación de nuevos atributos

- Añadir nuevos atributos aplicando expresiones a atributos ya existentes en el conjunto de datos.

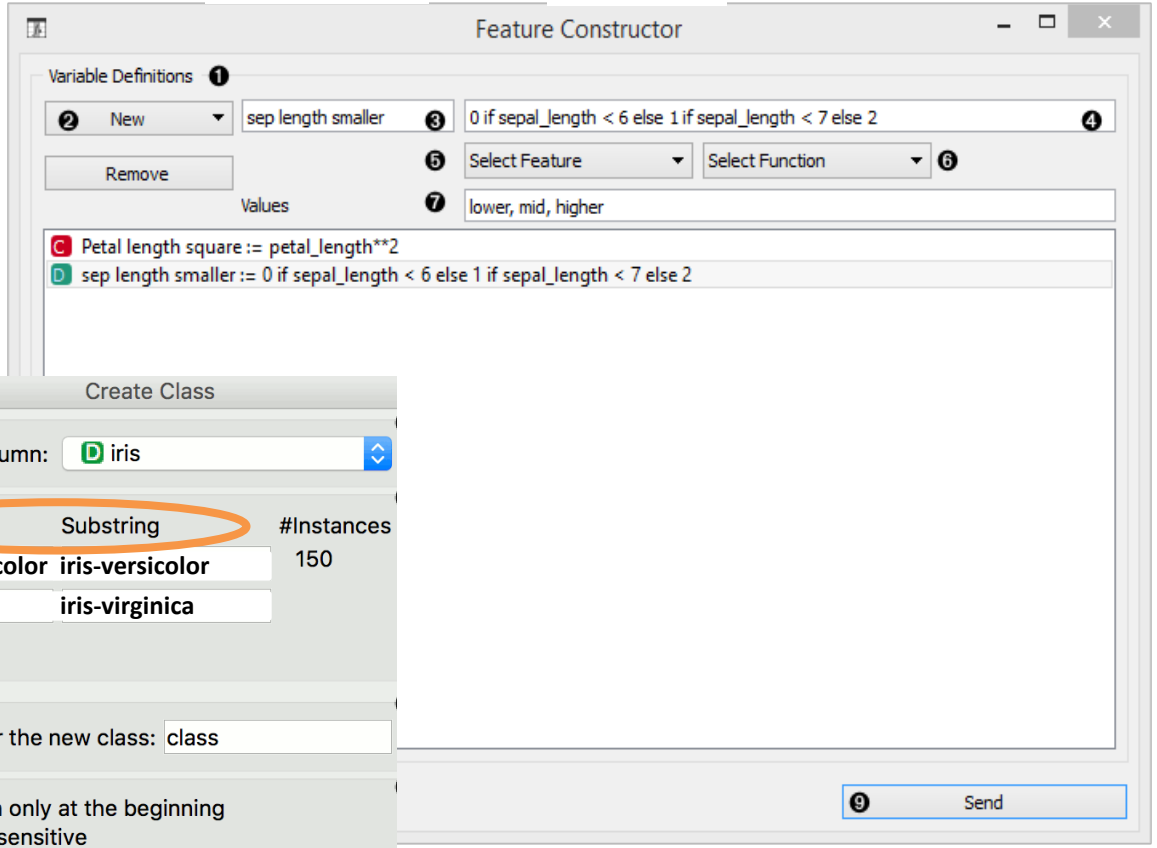
- Se pueden crear tanto atributos cuantitativos (por ejemplo usando expresiones en Python) como cualitativos (con sus posibles valores)

- **Nodo Feature Constructor**

- **Nodo Create Class**: Crea un nuevo atributo a partir de un atributo discreto o de texto (indicándolo en “Substring”)

Feature Constructor

Create Class



V. Normalización y estandarización

1. **Normalización:**
   - Trasforma el rango de valores a un intervalo determinado (normalmente [0,1]).
   - Es necesario si se van a aplicar algoritmos de aprendizaje basados en distancias para que todos los atributos estén en el mismo rango.

$$x_i^N = \left\{ \frac{x_{i,j} - \min(x_i)}{\max(x_i) - \min(x_i)}, \forall j \in 1,\ldots,n \right\}$$

$$x = x^N \cdot (\max(x_i) - \min(x_i)) + \min(x_i)$$

2. **Estandarización:**
   - Transforma los valores de los atributos para que tengan media 0 y desviación 1.
   - Sólo si la distribución es normal.

$$x_i^S = \left\{ \frac{x_{i,j} - \text{media}(x_i)}{\text{desv}(x_i)}, \forall j \in 1,\ldots,n \right\}$$

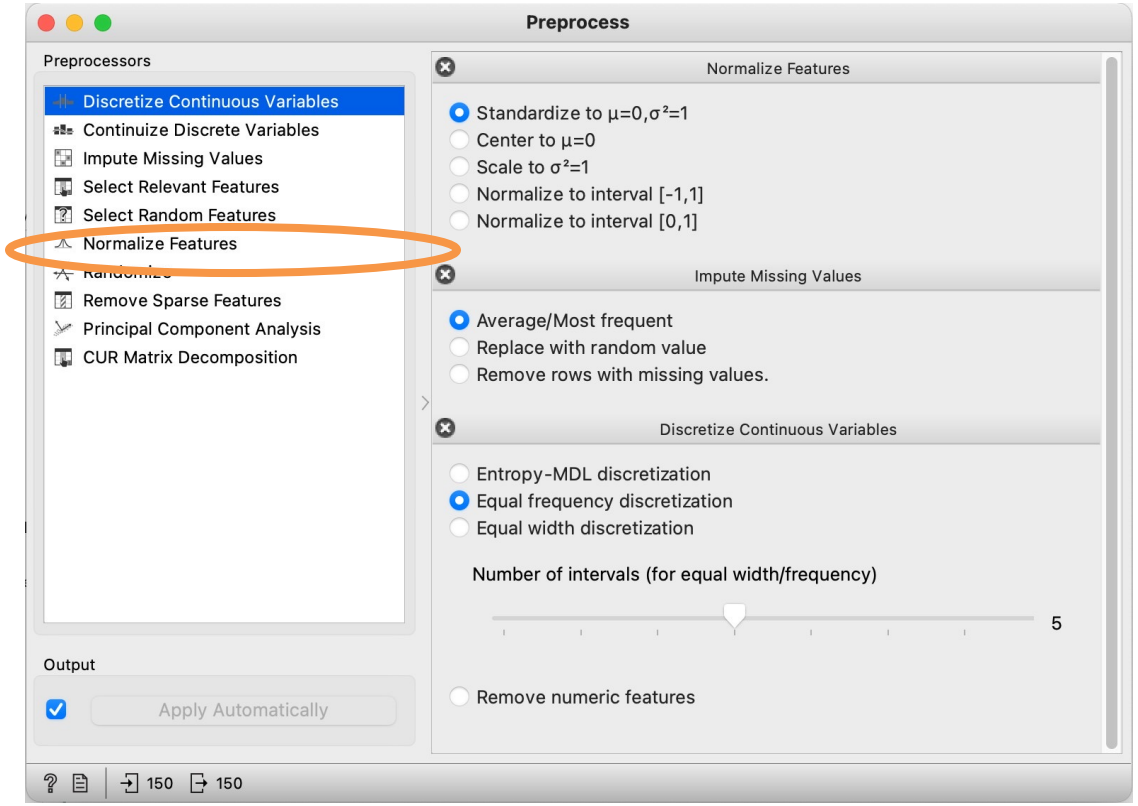
$$x = x^S \cdot \text{desv}(x_i) + \text{media}(x_i)$$

V. Normalización y estandarización

Nodo Preprocess

Normalize Features

- Ajustar valores a una escala común. Normalización y estandarización.



VI. Discretización de atributos continuos

Algoritmos de aprendizaje que solo operan con atributos discretos:

- Requieren transformación de atributos continuos a discretos.

Consiste en dividir el rango de valores continuos en un conjunto finito de intervalos (o cortes).

Opciones:

- No Supervisado:
  - Igual Anchura: k intervalos de igual anchura
  - Igual Frecuencia: k intervalos de N/k valores
- Supervisado:
  - Fayyad e Irani (basado en criterio de entropía mínima), Kononenko, 1Rules (Clasificación)

VI. Discretización de atributos continuos

No Supervisado:

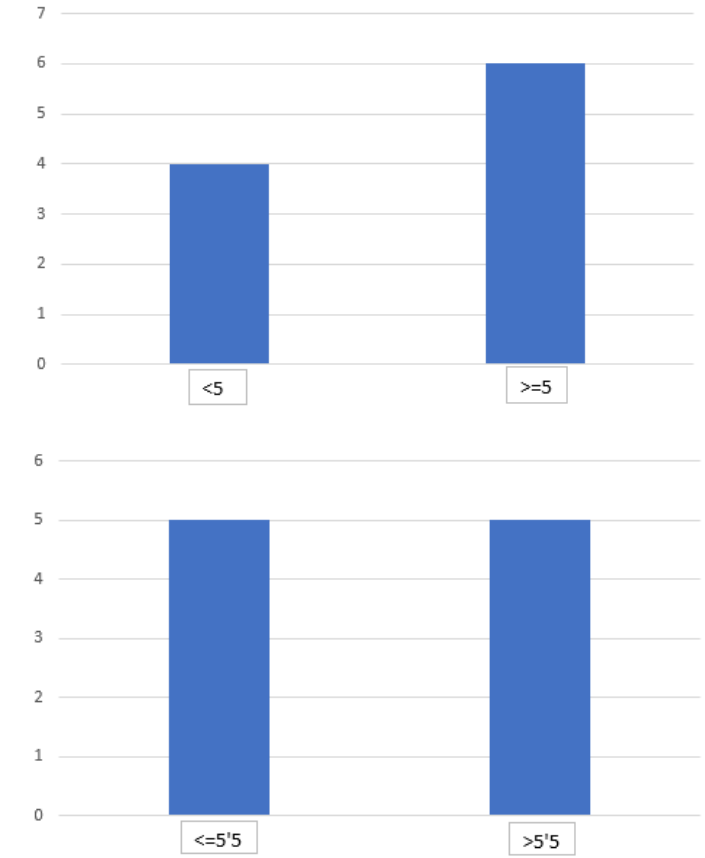
Ejemplo – Datos: 1, 2, 2’5, 4, 5’5, 6, 7, 7, 7’5, 9

Igual anchura

- Intervalo 1 (<5): 1, 2, 2’5, 4 [0, 5)
- Intervalo 2 (>=5): 5’5, 6, 7, 7, 7’5, 9 [5, 10]

Igual frecuencia

- Intervalo 1: 0, 2, 2’5, 4, 5’5 [0, 5’5]
- Intervalo 2: 6, 7, 7, 7’5, 9 (5’5, 10]

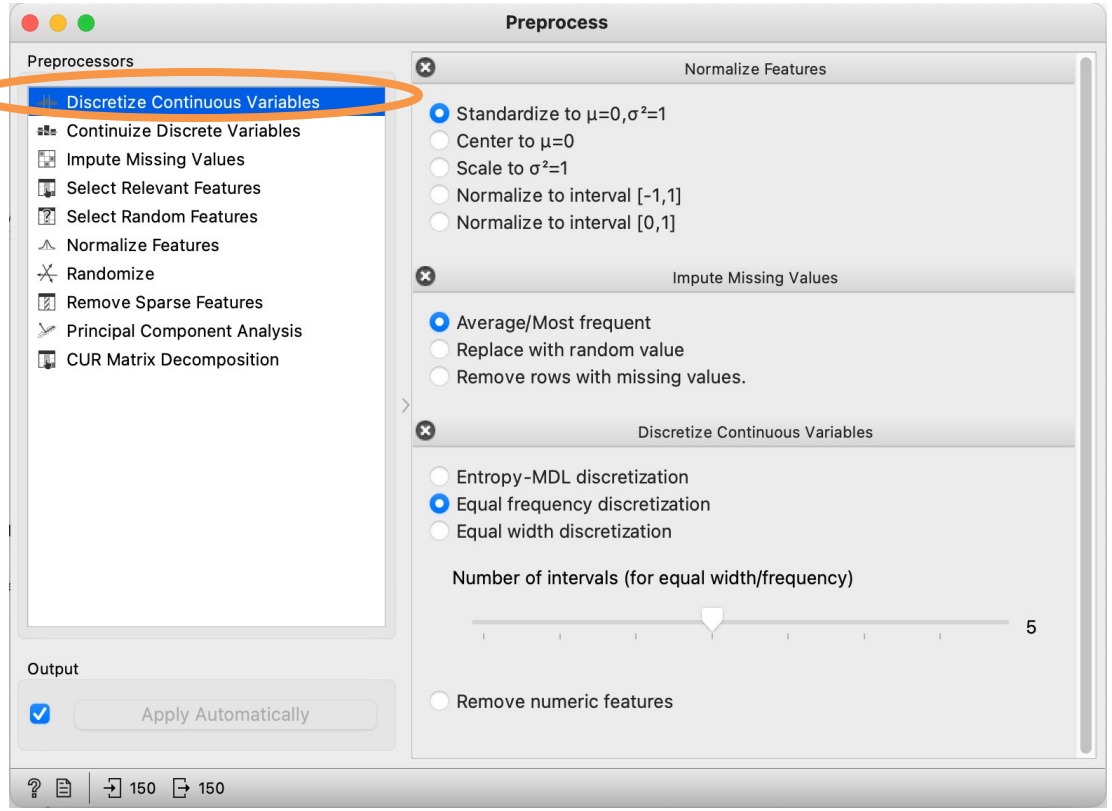


VI. Discretización de atributos continuos

Nodo Preprocess

Discretize Continuous Variables

- “Entropy-MDL discretization”: Método supervisado basado en el modelo de Fayyad and Irani para determinar el número de intervalos óptimos.
- “Equal frequency discretization”: Divide dejando el mismo número de instancias en cada intervalo.
- “Equal width discretization”: Crea intervalos con la misma anchura.
- “Remove numeric features altogether”: no recomendado, elimina todos los atributos numéricos

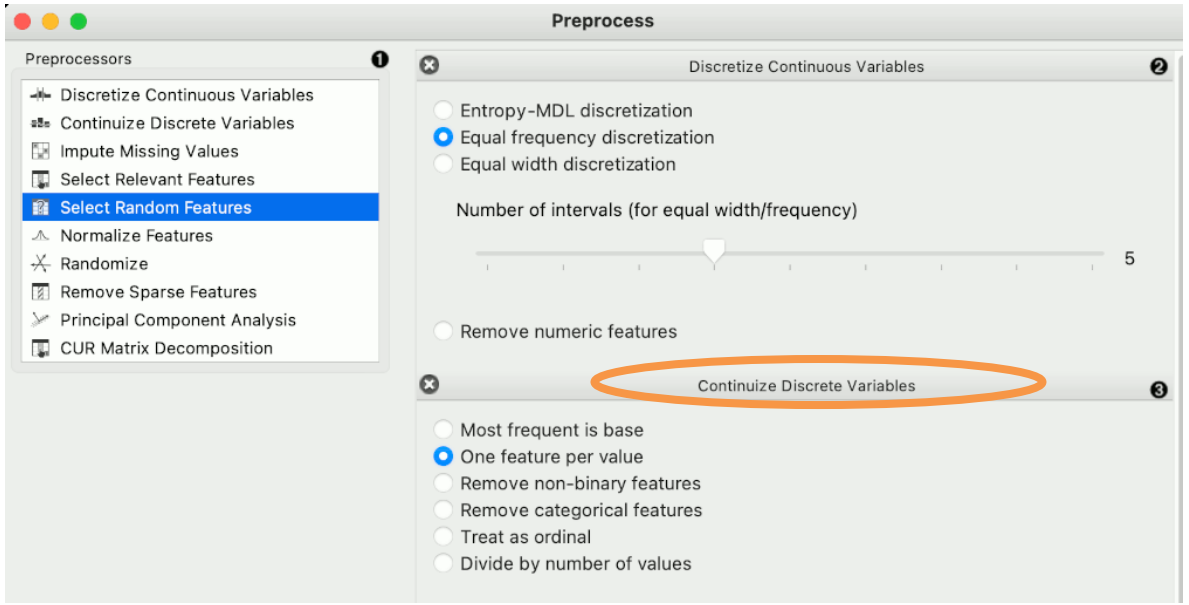


VII. “Continuización” de atributos discretos

- Es el proceso contrario a la discretización.
- Las variables discretas (incluyendo las binarias) se reemplaza por variables continuas.

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1



VII. “Continuización” de atributos discretos

Nodo Preprocess

Continuize Discrete Variables

- “Most frequent as base”: el valor más frecuente se codifica como 0 y los demás como 1.
- “One feature per value”: crea columnas para cada valor y pone 1 donde la instancia tenga ese valor (One Hot Encoding).

- “Remove non-binary features”: se queda solo con las características categóricas que tienen valores de 0 o 1 y las transforma en continuas.
- “Remove categorical features”: elimina todos los atributos categóricos.

- “Treat as ordinal”: toma valores discretos y los trata como números. Si los valores discretos son categorías, cada categoría será asignada a un número en el orden en el que aparecen en los datos.
- “Divide by number of values”: es similar al anterior, pero el número final será dividido por el número total de valores, de esta forma el rango de la nueva columna estará comprendido entre [0,1].

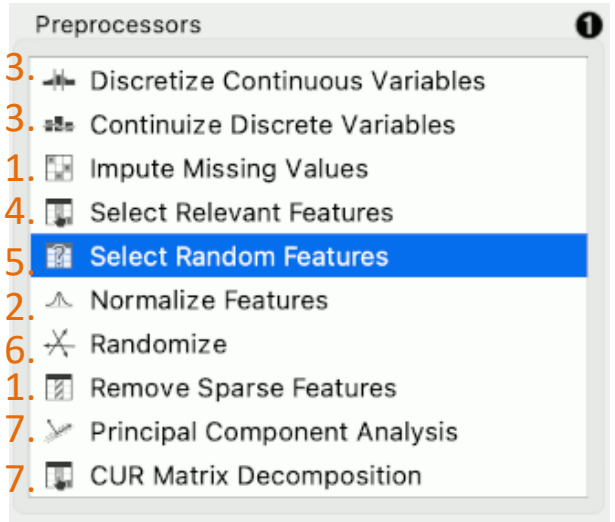
VIII. Preprocesado en Orange

- Las herramientas de preprocesado en Orange se encuentran en el modulo **Data**.

- Orange contiene un nodo específico para preprocesamiento **Preprocess** en el que se puede crear un pipeline de preprocesado pulsando y arrastrando componentes.

Ya estudiados:

1. Tratamiento de valores ausentes
2. Normalización y estandarización
3. Discretización y continuización



Más tipos:

4. Selección de atributos relevantes
5. Selección al azar de atributos
6. Aleatorizar instancias: Mezcla los valores de las clases y destruye cualquier conexión existente entre las instancias y las clases
7. Métodos para la reducción de la dimensionalidad: PCA y CUR

IX. Actividad 1: Valores ausentes y outliers

1. **Filtros de valores ausentes:**
   a) Cargue el fichero "**diabetes_missing.csv".** Es un conjunto de datos con valores ausentes. Las variables o atributos que se miden de cada instancia está en:
   https://github.com/gualbe/datasets/blob/main/curso_ciencia_datos/diabetes_missing.csv
   Todos los atributos son numéricos menos la clase que es el atributo categórico "Outcome". Realice la importación del fichero indicando los tipos de cada atributo.

   b) Orange detecta los valores ausentes con **?** . ¿Cuál es el porcentaje de atributos ausentes en el dataset?

   c) A continuación, veamos cómo podemos evitar los valores ausentes de nuestro conjunto de datos. Una forma sencilla de manejar los datos que faltan es quitar aquellas instancias que tienen uno o más valores ausentes. Podemos hacerlo en Orange utilizando el nodo **Preprocessing** o **Impute**:
      - Seleccione el filtro **Impute**.
      - Establezca el método por defecto **Remove Instances with unknown values.**
      - Ejecute y observe si se han eliminado las instancias indicadas visualizando el conjunto de datos transformado. ¿Cuántas instancias tiene el nuevo conjunto de datos?

IX. Actividad 1: Valores ausentes y outliers

d) Ahora en lugar de eliminar instancias con valores perdidos (no recomendado), vamos a reemplazarlos con otros valores. En este caso vamos a tratar los valores perdidos. Para hacer esto, vamos a usar el filtro **Impute** de nuevo. Es común tratar los valores perdidos con la media de la distribución numérica.

- Seleccione la opción **Average/most frequent**. Ejecute y observe si se han reemplazado los valores indicados visualizando el conjunto de datos transformado.

2. Detección de Outliers:

a) Por último, vamos a observar si hay valores que no sigan la tendencia normal de los datos en el conjunto de datos **diabetes.**

b) Para hacernos una idea de si habrá o no outliers primero vamos a utilizar un box plot. ¿Qué conclusiones puede sacar?

c) Ejecute el nodo **Outliers** con el método "Local Outlier Factor" con 20 vecinos y la distancia euclídea.

d) Conecte la salida del nodo con dos Data Table, uno para los outliers y otro para los inliers (seleccionándolo en el canal). ¿Cuántos outliers se han detectado?

IX. Actividad 2: Preprocesado en IRIS

1. **Cargue el dataset IRIS**

2. **Comprobación de datos**
   a) Utilice el nodo **Select Columns** y compruebe que los atributos y la clase ("iris") están seleccionados de forma correcta.
   b) Visualice los datos con **Data Table**.

3. **Discretización**
   a) Utilice el filtro **Discretize** (no supervisado) sobre el conjunto de datos iris. Establezca la propiedad bins a 4 usando intervalos de misma anchura. Visualice el conjunto de datos tras la transformación usando el nodo **Distributions**.

   b) Repita el mismo filtro pero ahora divida el dominio de los atributos en 4 intervalos de la misma frecuencia. Visualice el conjunto de datos tras la transformación y compare los resultados obtenidos con el primer filtro.

   c) Repita el filtro con la propiedad Entropy-MDL discretization a True (**Supervisada**). Visualice el conjunto de datos tras la transformación y compare los resultados obtenidos con el primer filtro.

IX. Actividad 2: Preprocesado en IRIS

4. Filtros de transformación: Atributos calculados, normalización y binarización

a) Añada dos atributos calculados mediante el nodo **Feature Constructor**: uno que multiplique los atributos relacionados con el sépalo y otro que multiplique los relacionados con el pétalo. Visualice el conjunto de datos tras la transformación realizada con **Scatter Plot**. Analice los resultados obtenidos.

b) Aplique el filtro **Normalize** en el intervalo [0,1] del nodo **Preprocess** sobre el conjunto de datos generado en el paso anterior. Visualice los resultados con **Scatter Plot**. ¿Qué efecto tiene este filtro?

c) Transforme los valores de los atributos para que tengan media 0 y desviación 1 utilizando el filtro **Standarize** del nodo **Preprocess**.

5. Selección de instancias:

a) Aplique el nodo **Select Rows**, que selecciona las instancias de acuerdo a condiciones definidas sobre uno de los atributos. En este caso se desean eliminar las instancias cuya clase sea igual a **Iris-setosa**. Repita el proceso, pero ahora obtenga aquellas instancias con valor superior a 5.2 en el atributo **sepalLength**.

b) Aplique el filtro **DataSampler** para escoger el 80% de las instancias. Guarde el conjunto de datos resultante como **iris2.csv**