Módulo 8

8.3 Aprendizaje Supervisado: Regresión y Clasificación en KNIME

Por **María Martínez Rojas**
Profesora Titular en CA, Universidad de Granada

Por **José Manuel Soto Hidalgo**
Profesor Titular en ICAR, Universidad de Granada

1. INTRODUCCIÓN

Esta cápsula se centra en la implementación en KNIME de los distintos algoritmos de aprendizaje supervisado que se han introducido en los módulos 4 y 5. Se realizarán flujos de datos representativos del ciclo de vida de Ciencia de Datos para resolver problemas de regresión lineal (4.2) y con árbol (4.3) problemas de clasificación con métodos básicos como son el vecino más cercano (KNN) y los árboles de decisión (5.2), así como con métodos avanzados como Random Forest (5.3). Se utilizarán también procesos de validación de los modelos para dar soporte estadístico a los resultados obtenidos.

2. REGRESIÓN EN KNIME

En este apartado se van a crear flujos de datos para resolver un problema de regresión con un regresor lineal (Módulo 4, cápsula 2) y un árbol (Módulo 4, cápsula 3). Se va a utilizar el conjunto de datos HOMA y se ilustrarán distintos elementos en KNIME para visualizar qué variables son más prometedoras para aplicar la regresión, dibujar gráficas con la regresión, así como distintas métricas de calidad de los modelos de regresión.

2.1. ¿CÓMO RESOLVER UN PROBLEMA DE REGRESIÓN LINEAL?

En este flujo de datos se va a utilizar un nodo *Linear Regression Learner* y un nodo *Regression Predictor* para realizar la regresión lineal (Figura 1).

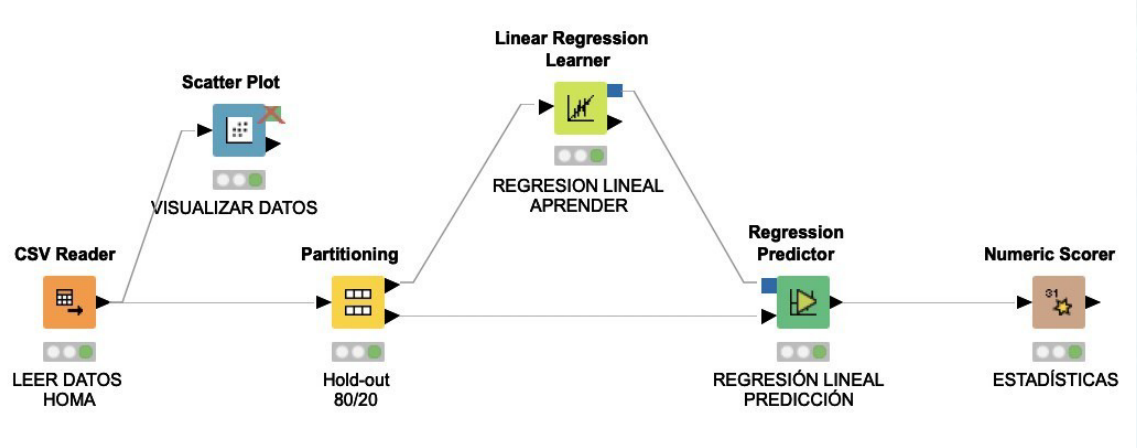


Figura 1 Flujo de datos Regresión Lineal

Inicialmente, se lee el conjunto de datos HOMA con un nodo *CSV Reader*. Una vez leídos los datos, un elemento interesante en un problema de regresión como primera toma de contacto consiste en analizar visualmente las variables y observar correlación entre ella. Un nodo que permite realizar este aspecto es el nodo *Scatter Plot*, el cual permite generar un gráfico 2D interactivo con el que poder visualizar los datos asociados a dos variables. Por ejemplo, podemos visualizar distintas las variables del conjunto de datos con respecto a HOMA. Para ello, hacemos click en el icono de menú de la parte superior derecha del gráfico, y seleccionamos las variables que queremos representar en el eje X e Y. La Figura 2 y Figura 3 muestran gráficamente la dispersión de las variables Sex y SBP respecto a HOMA, respectivamente.

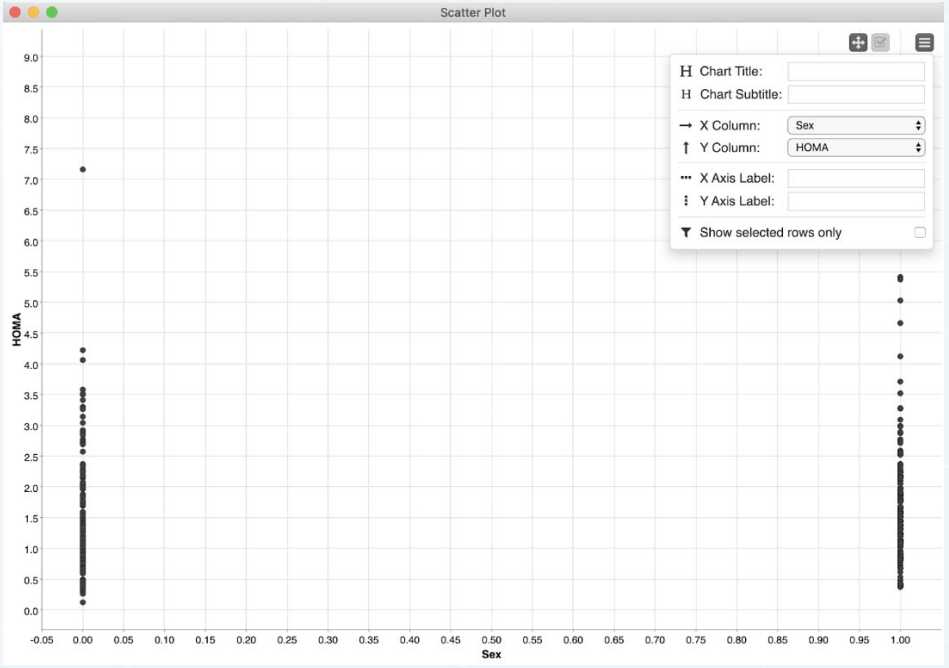


Figura 2: Flujo de datos Regresión Lineal – Visualización de la variable Sex respecto a HOMA

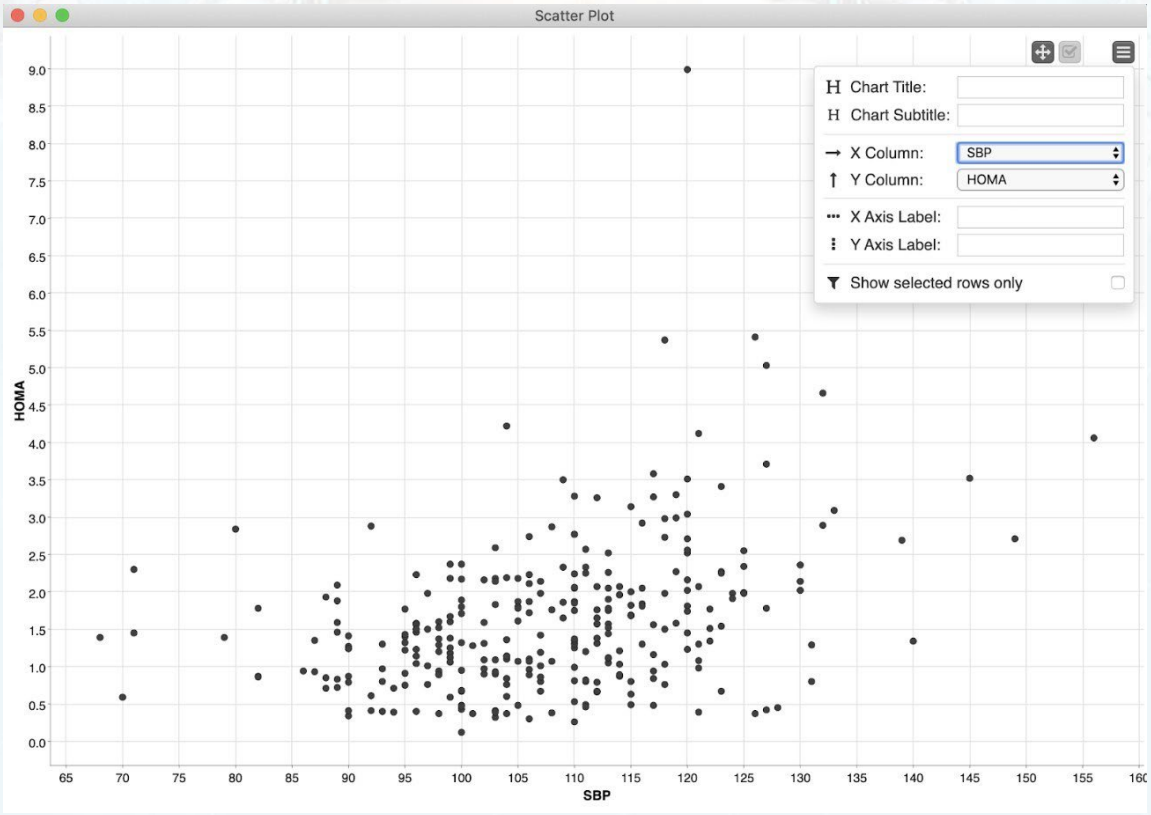


Figura 3: Flujo de datos Regresión Lineal – Visualización de la variable SBP respecto a HOMA

Una vez leídos los datos, éstos se usan como entrada a un nodo de *Partitioning*, para obtener un conjunto de datos de test y entrenamiento del modelo mediante el procedimiento hold-out. Con este nodo se puede configurar el porcentaje de la primera partición de forma absoluta o relativa así como el tipo de particionamiento: desde la parte superior, muestreo lineal o aleatorio. Además, se puede fijar una semilla para que el particionamiento aleatorio siga el mismo patrón en distintos ejemplos. La Figura 4 muestra el diálogo de configuración y parámetros del nodo *Partitioning*.

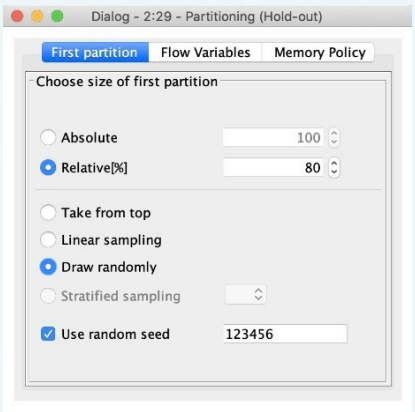


Figura 4: Flujo de datos Regresión Lineal – Opciones y parámetros de Partitioning

El nodo *Linear Regression Learner* utiliza los datos de entrenamiento para crear el modelo. La Figura 5 muestra las opciones de configuración del nodo *Linear Regression Learner*, donde se puede seleccionar la variable objetivo, en este ejemplo la variable HOMA, así como las distintas variables a incluir en el proceso de aprendizaje del modelo regresor. También se puede indicar el tratamiento que se quiere hacer a los valores perdidos.

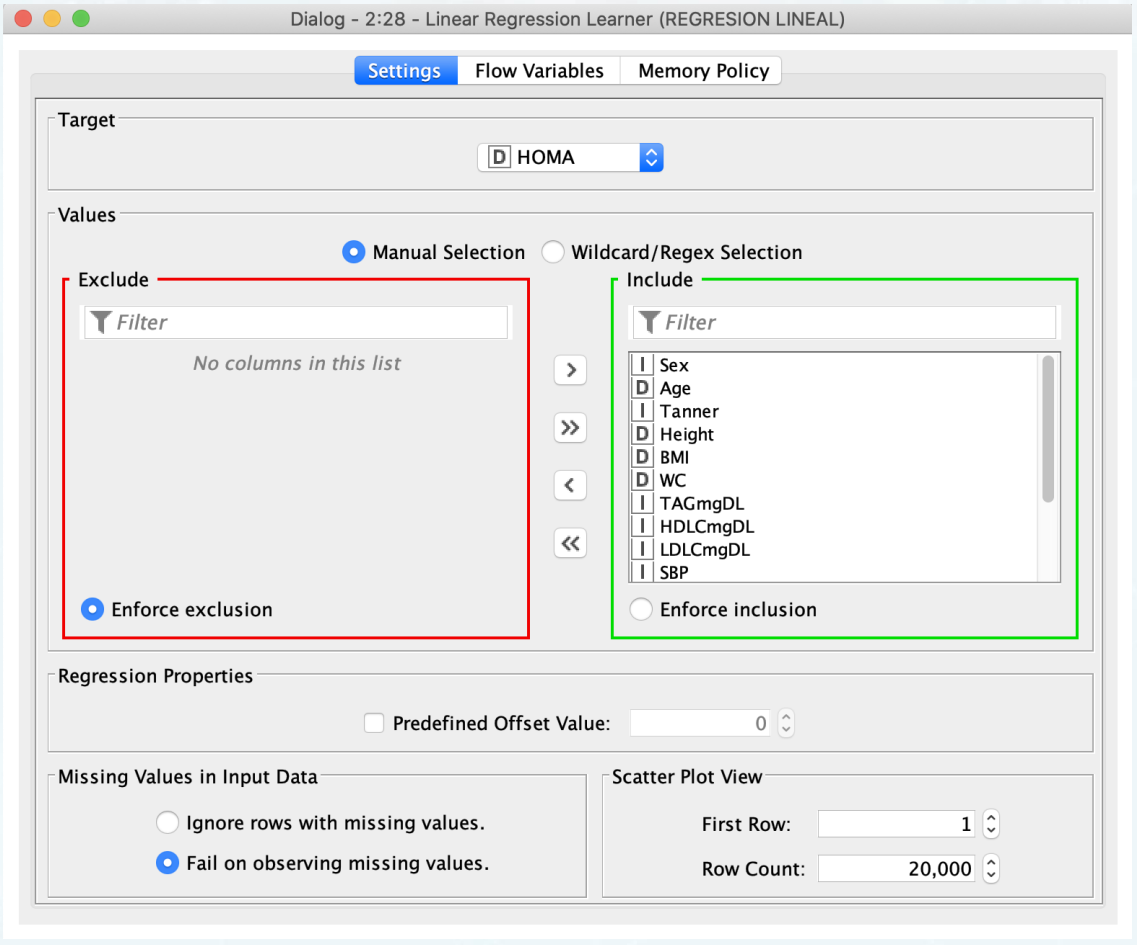


Figura 5: Flujo de datos Regresión Lineal – Nodo Linear Regression Learner

La salida del nodo *Linear Regression Learner* son dos puertos, uno correspondiente al modelo regresor y otro con los coeficientes y datos estadísticos del modelo regresor. Los coeficientes y datos estadísticos se pueden visualizar, una vez ejecutado el nodo (semáforo en verde), haciendo click con el botón derecho en el nodo y en la opción del menú desplegable "Linear Regression Results View" donde se visualizan las estadísticas del modelo, como los coeficientes, error estándar, t-value y P>|t| para cada una de las variables del conjunto de datos (Figura 6).

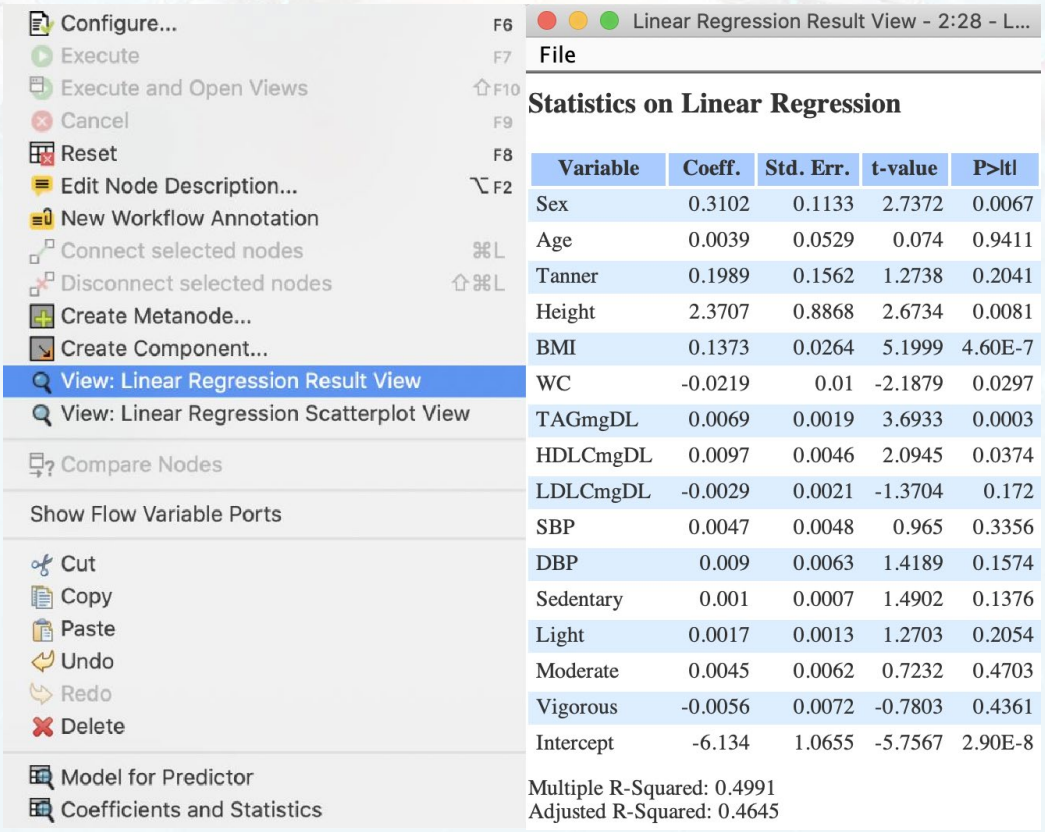


Figura 6: Flujo de datos Regresión Lineal – Resultados del modelo regresor

También se pueden visualizar gráficamente los valores estimados por el modelo lineal frente a los valores reales a través del menú desplegable “Linear Regression Scatterplot View”. Por ejemplo, la Figura 6 muestra gráficamente los valores estimados (línea recta) de la variable Height frente a los valores reales.

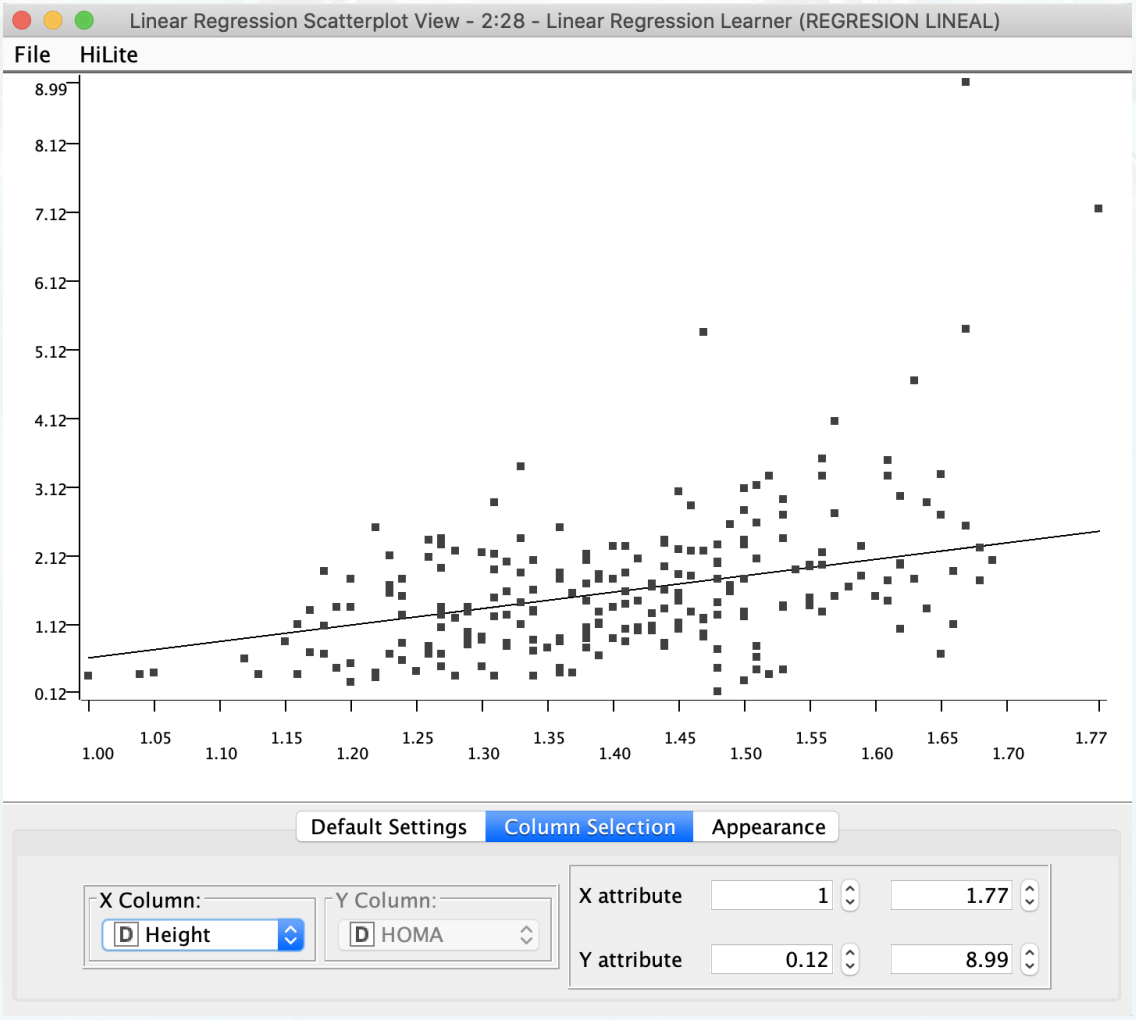


Figura 7: Flujo de datos Regresión Lineal – Valores estimados vs valores reales de la variable Height

Finalmente, el nodo *Regression Predictor*, recibe por un puerto el modelo regresor y los datos de test para evaluar la predicción. La salida de este son el conjunto de datos al que se añade la predicción. A partir de estos datos se pueden obtener estadísticas como métrica de la calidad de la predicción con el nodo *Numeric Scorer*. La Figura 8 muestra algunas métricas como R^2, error absoluto medio, error cuadrático medio, etc.

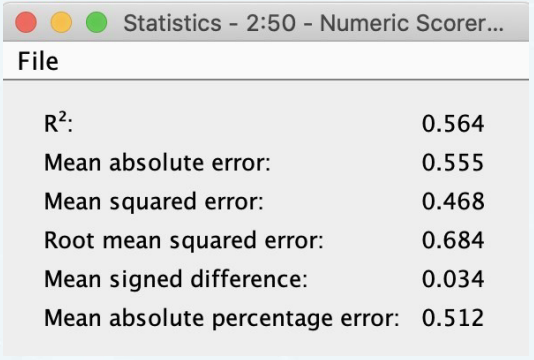


Figura 8: Flujo de datos Regresión Lineal – Resultados estadísticos

2.2. ¿CÓMO RESOLVER UN PROBLEMA DE REGRESIÓN CON UN ÁRBOL?

En este flujo de datos se va a utilizar un nodo *Simple Regression Tree Learner* y un nodo *Simple Regression Tree Predictor* para realizar la regresión (Figura 1). Nótese que para incluir el nodo que implementa el algoritmo de regresión M5 utilizado en la cápsula 3 del módulo 4 es necesario instalar una extensión en KNIME. Como el objetivo de este apartado es ilustrar cómo resolver un problema de regresión con un árbol, se va a utilizar el nodo *Simple Regression Tree Learner* que representa el concepto de regresión con árbol a través del ampliamente conocido algoritmo CART.

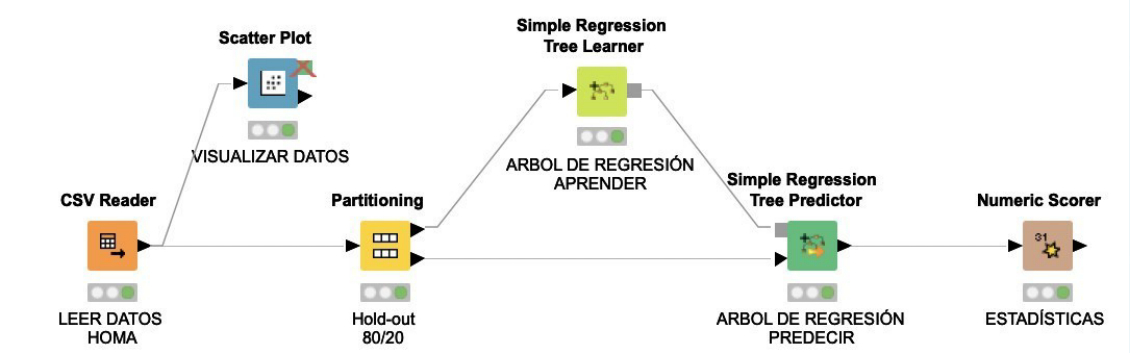


Figura 9: Flujo de datos Regresión Árbol

Al igual que en el ejemplo anterior, se leen los datos del conjunto de datos HOMA, se particionan con un hold-out de 80% para entrenamiento y un 20% para test, los cuales se van a utilizar para aprender el modelo y en consecuencia como entrada al nodo *Simple Regression Tree Learner* y para test como entrada al nodo *Simple Regression Tree Learner* respectivamente La Figura 10 muestra las opciones y parámetros de configuración del nodo *Simple Regression Tree Learner*.

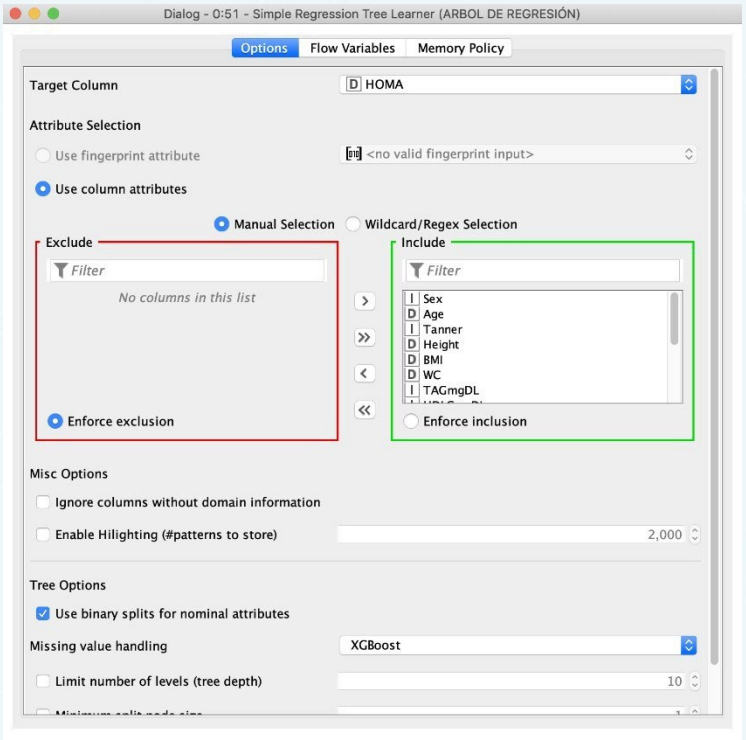
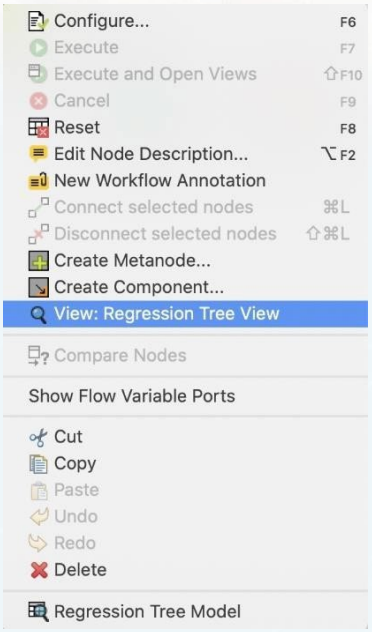


Figura 10: Opciones y parámetros de configuración de Regresión Árbol

Una vez ejecutado el nodo *Simple Regression Tree Learner* éste aprende un modelo de árbol el cual se puede visualizar a través de la opción del menú View Regression Tree View (Figura 11).



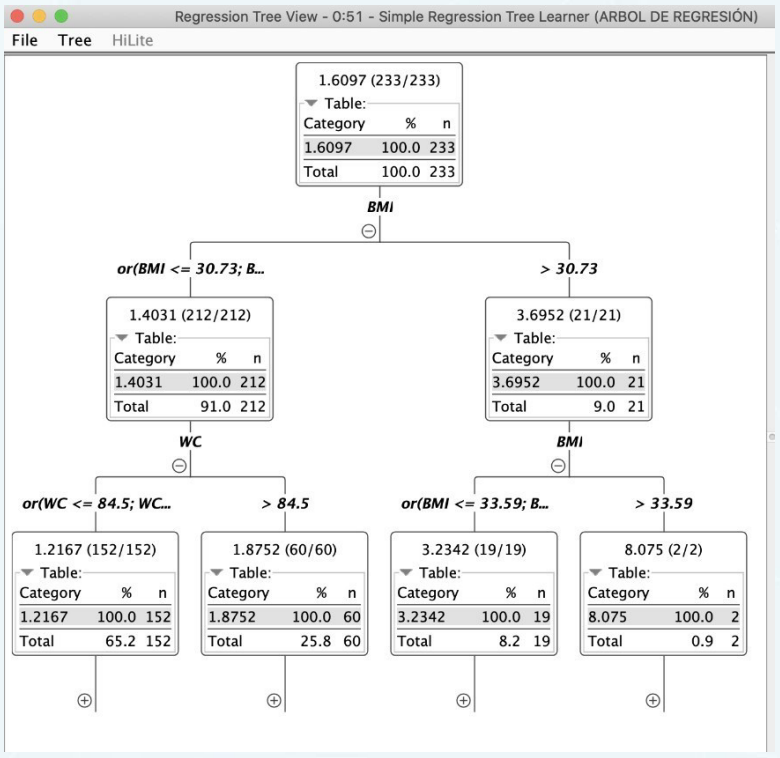


Figura 11: Visualización del Regresión Árbol

Finalmente, al igual que en el ejemplo anterior, el modelo aprendido con el nodo *Simple Regression Tree Learner* se utiliza para predecir con el nodo *Simple Regression Tree Predictor* conectado a través del puerto de entrada (el modelo regresor) y los datos de test para evaluar la predicción. A partir de estos datos se pueden obtener estadísticas como métrica de la calidad de la predicción con el nodo *Numeric Scorer*. La Figura 12 muestra algunas métricas como R^2, error absoluto medio, error cuadrático medio, etc.

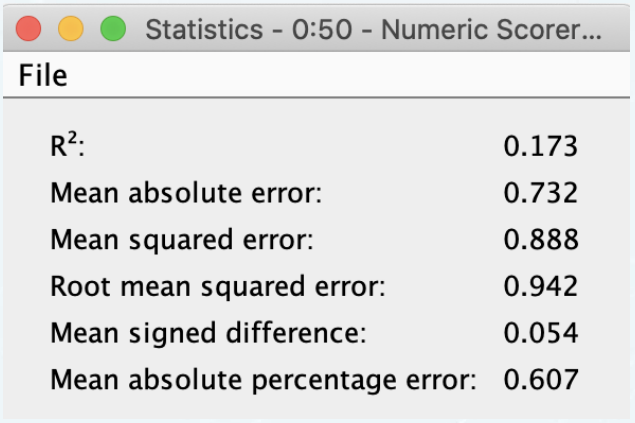


Figura 12: Flujo de datos Regresión Árbol– Resultados estadísticos

3. CLASIFICACIÓN EN KNIME

En este apartado se van a crear flujos de datos para resolver un problema de clasificación con los algoritmos KNN (sección 3.2.1), árbol de decisión (sección 3.2.2) y Random Forest (sección 3.2.3) considerando como datos el conjunto de datos de expresión genética para el problema de melanoma cutáneo mencionado en el módulo 5. Finalmente se realizará un ejemplo de cómo comparar distintos algoritmos a través de una curva ROC (sección 3.2.4). En cada flujo de datos se van a integrar distintos elementos de KNIME para ilustrar la utilidad y versatilidad de KNIME a la hora de resolver problemas de Ciencia de Datos.

3.1 ¿CÓMO RESOLVER UN PROBLEMA DE CLASIFICACIÓN CON KNN?

En este flujo de datos se va a utilizar el algoritmo KNN (Figura 13). Como en todos los ejemplos, primero se leen los datos del conjunto de datos inmune con los nodos *CSV Reader*. Con el nodo *Column Appender* se unen en una sola tabla de datos los datos procedentes de los datos de variables (inmune_X) y los datos asociados a la clase (inmune_Y). Esos datos se usan como entrada a un nodo de *Partitioning*, para obtener un conjunto de datos de test y entrenamiento del modelo. El nodo *K Nearest Neighbor* utiliza los datos de entrenamiento y de test para crear el modelo y mostrar los resultados de clasificación. Para visualizar los resultados de clasificación se ha utilizado un nodo *Scatter Plot* y para

medir la calidad del modelo obtenido para la clasificación, se ha usado un nodo *Scorer* donde se puede observar la matriz de confusión y las estadísticas (accuracy, f-measure, etc.).

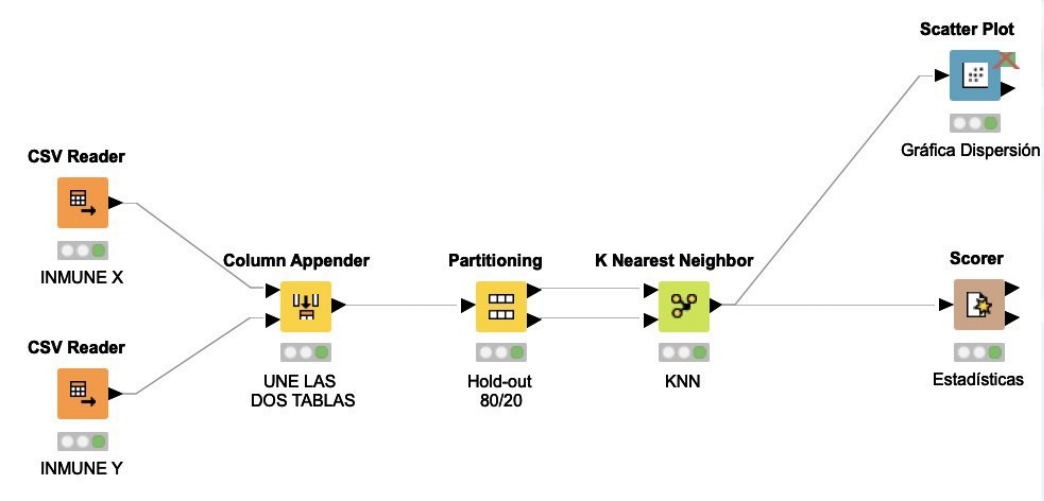


Figura 13: Flujo de datos KNN

El modelo de aprendizaje automático en este ejemplo es el nodo *K Nearest Neighgor* el cual tiene los parámetros que se indican a continuación: La columna de la tabla de datos que contiene la clase a clasificar, el número de vecinos a considerar, etc. (Figura 14).

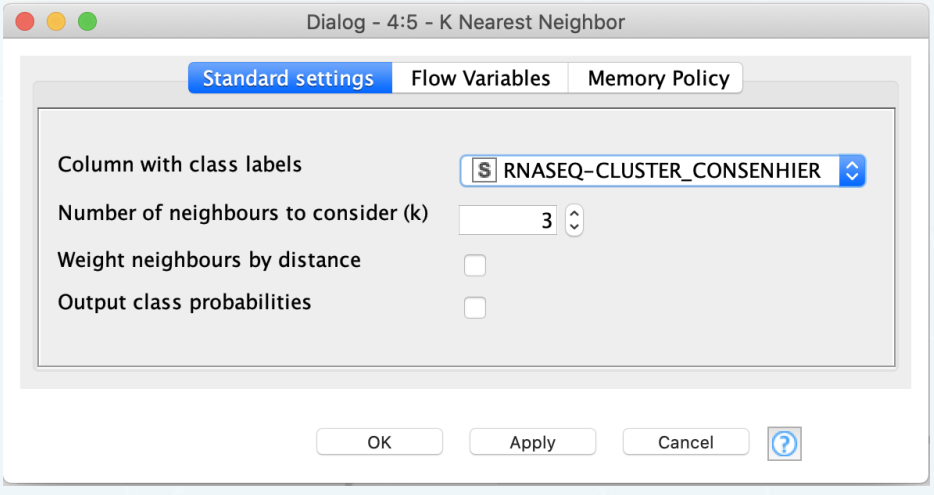


Figura 14: Opciones y parámetros de configuración de KNN

También se ha utilizado un nodo *Scatter Plot*, el cual permite visualizar datos de forma interactiva en dos dimensiones. Se pueden configurar las variables a representar en el eje X e Y. La Figura 15 ilustra un ejemplo de visualización. Por último, el nodo *Scorer* se pueden obtener la matriz de confusión (Figura 16) así como información sobre distintos indicadores de calidad del modelo, como accuracy, precisión, recall, falsos positivos, etc. (Figura 17).

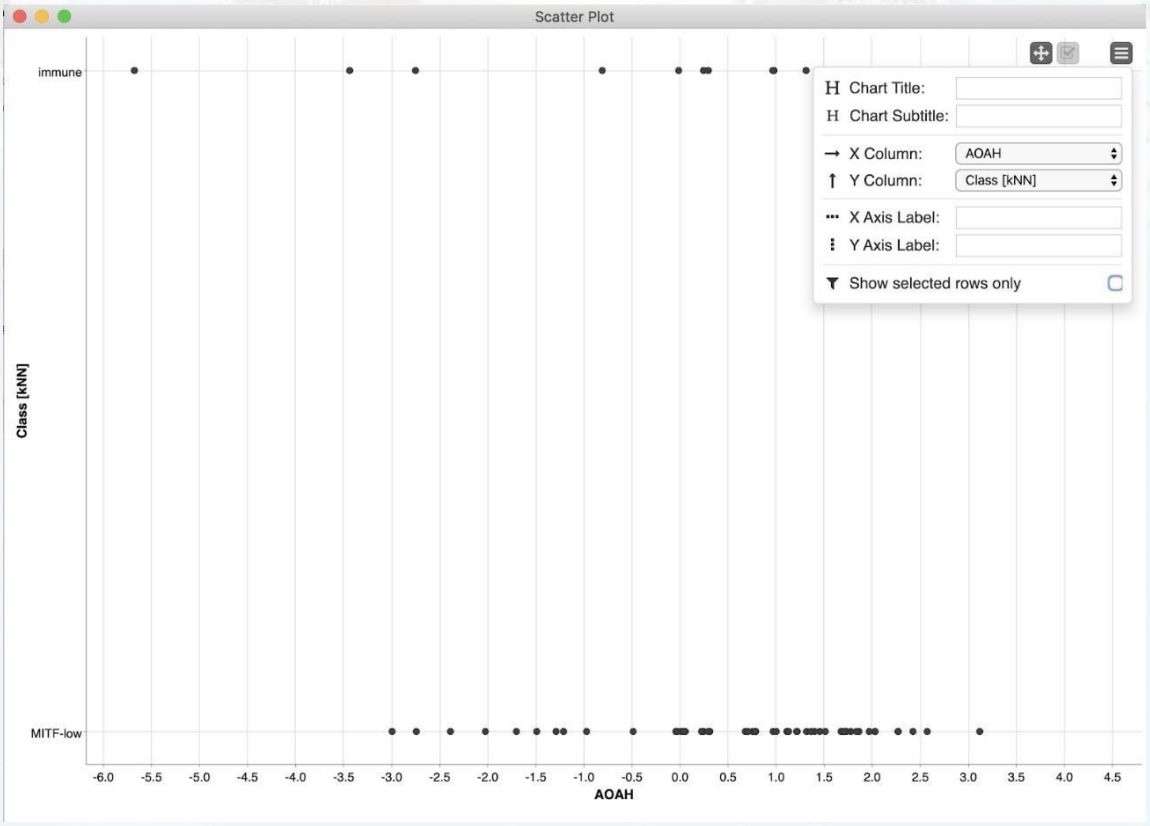


Figura 15: Flujo de datos KNN – Scatter Plot

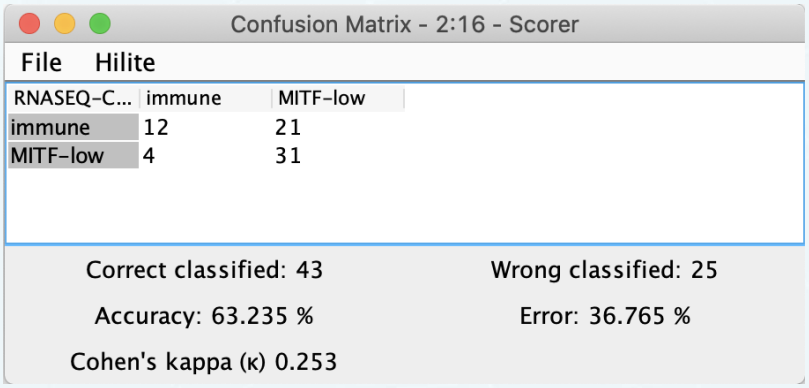
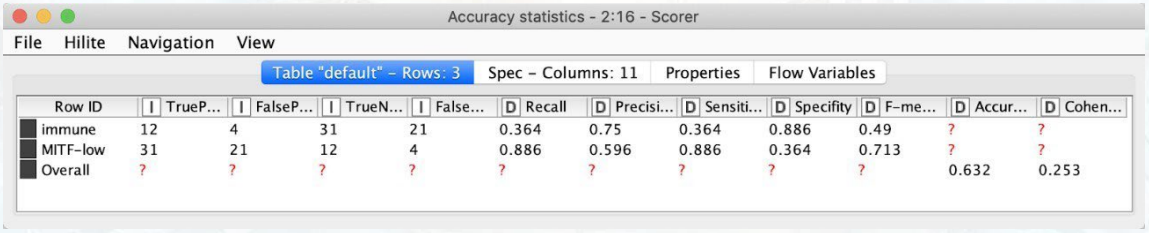


Figura 16: Flujo de datos KNN – Matriz de confusión



Row ID	TrueP...	FalseP...	TrueN...	False...	Recall	Precisi...	Sensiti...	Specifity	F-me...	Accur...	Cohen...
immune	12	4	31	21	0.364	0.75	0.364	0.886	0.49	?	?
MITF-low	31	21	12	4	0.886	0.596	0.886	0.364	0.713	?	?
Overall	?	?	?	?	?	?	?	?	?	0.632	0.253

Figura 17: Flujo de datos KNN - Tabla con resultados estadísticos

3.2 ¿CÓMO RESOLVER UN PROBLEMA DE CLASIFICACIÓN CON UN ÁRBOL DE DECISIÓN?

En este flujo de datos se va a utilizar un Árbol de Decisión (Figura 18). Al igual que en el ejemplo anterior, primero se leen los datos del conjunto de datos inmune con los nodos CSV Reader, se agrupan en una sola tabla y se dividen con los nodos *Column Appender* y *Partitioning* respectivamente. En este caso, los datos de entrenamiento se utilizan como datos de entrada para el nodo *Decision Tree Learner* el cual aprende un modelo que es el que utiliza el nodo *Decision Tree Predictor* junto con los datos de test para obtener los resultados de clasificación. En este flujo se ha utilizado también un nodo *Decision Tree View* para ver de manera visual el árbol de decisión obtenido. De igual forma que en ejemplos anteriores, se usa un nodo *Scorer* para medir la calidad del modelo obtenido para la clasificación.

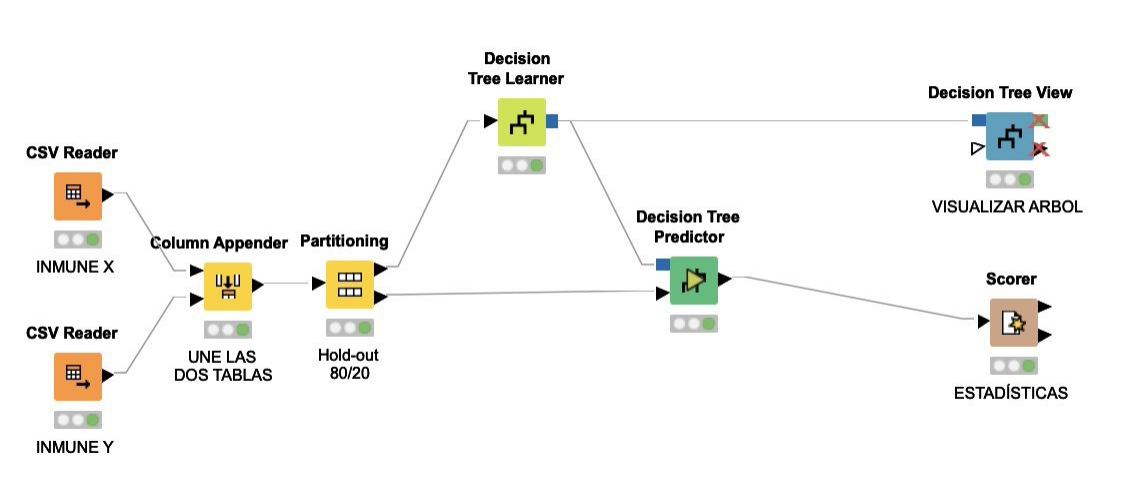


Figura 18: Flujo de datos Árbol de Decisión

El nodo *Decision Tree Learner* dispone de varios parámetros tal y como se detalla en el módulo 5 como la clase a la que clasificar, la medida de calidad, el método de poda, etc. La Figura 19 muestra los parámetros por defecto.

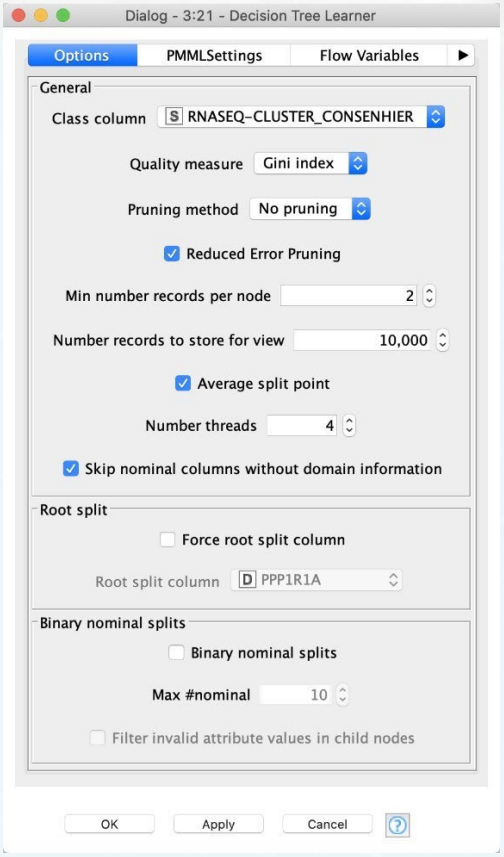


Figura 19: Opciones y parámetros de configuración del Árbol de Decisión

Un interesante nodo en KNIME es el nodo *Decision Tree View*, el cual permite visualizar el árbol de decisión de manera interactiva. La Figura 20 muestra un ejemplo del árbol aprendido.

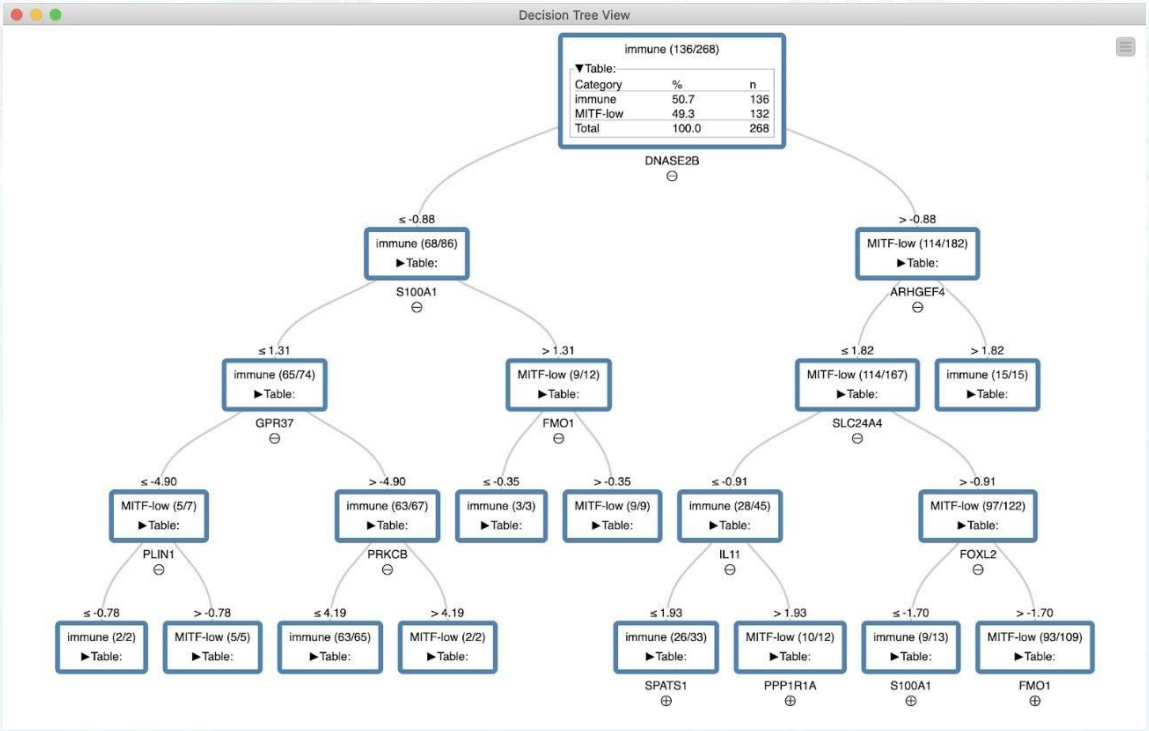


Figura 20: Flujo de datos Árbol de Decisión – Visualización del Árbol de Decisión

3.3 ¿CÓMO RESOLVER UN PROBLEMA DE CLASIFICACIÓN CON RANDOM FOREST?

En este flujo de datos se va a utilizar el algoritmo Random Forest (Figura 25). Como elemento a destacar en este flujo de datos con respecto a los anteriores es la utilización de un metanodo (detallado en la cápsula 1) para realizar la validación cruzada. La lectura y manipulación de datos es la misma que los ejemplos anteriores, pero el metanodo *Cross Validation* nos va a permitir incluir el algoritmo Random Forest y utilizar una validación cruzada.

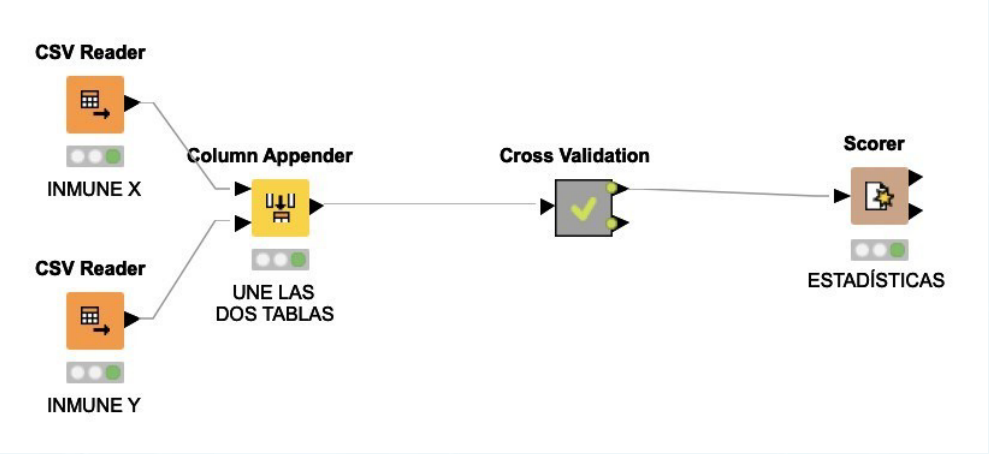


Figura 25: Flujo de datos Random Forest

La Figura 26 muestra un ejemplo del metanodo de validación cruzada.

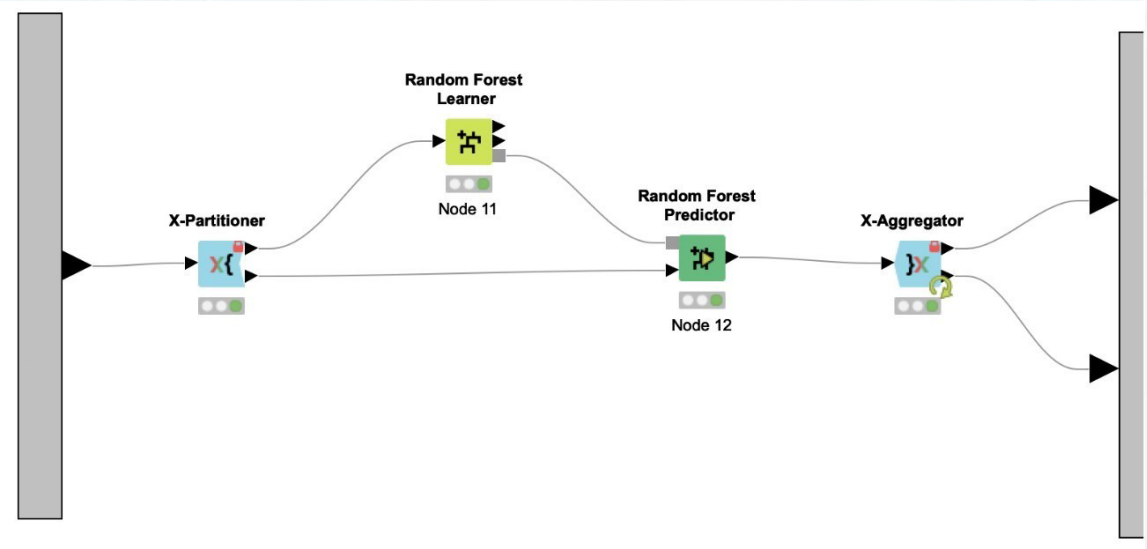


Figura 26: Flujo de datos Random Forest – Metanodo Cross Validation

En este ejemplo, se integran los nodos *Random Forest Lerarner* y *Random Forest Predictor* junto con los nodos *X-Partitioner* y *X-Aggregator*. Los nodos *Random Forest Learner* y *Random Forest Predictor*, se encargan de crear el modelo y predecir acorde al modelo, respectivamente, mientras que los nodos *X-Partitioner* y *X-Aggregator* se encargan de dividir los datos, y de agregar el resultado tantas veces como número de validaciones se hayan fijado en los parámetros, es decir, de realizar tantas iteraciones como número de validaciones se hayan fijado. La Figura 27 muestra las distintas opciones y parámetros que presenta el nodo *Random Forest Lerarner*, entre los que destacan la clase objetivo, los atributos a incluir en el proceso de aprendizaje, y el mecanismo de separación (Split Criterion) por defecto "Information Gain Ratio".

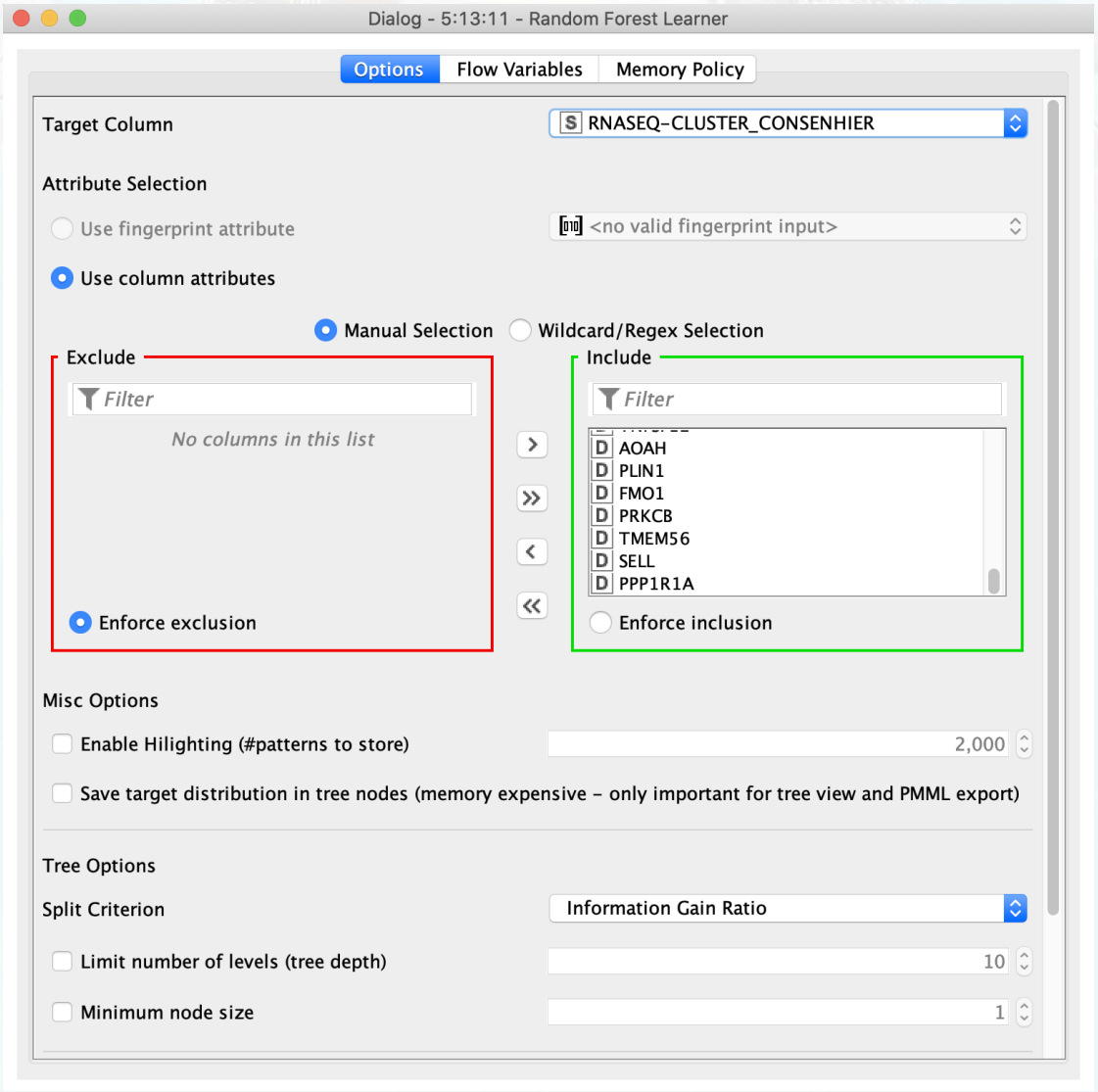


Figura 27: Opciones y parámetros del nodo Random Forest Learner

Las opciones y parámetros que presenta el nodo *X-Partitioner* son el número de validaciones (folds), el muestreo si es lineal, aleatorio o estratificado y la clase objetivo (Figura 28) mientras que el nodo *X-Aggregator* solo tiene como parámetros la columna que representa la clase objetivo y la predicción (Figura 29).

IMPORTANTE: Si se quiere realizar una experimentación, en la misma máquina, comparativa con varios algoritmos, es recomendable marcar la opción Random Seed y fijar una semilla para que siempre se utilicen los mismos datos en el proceso de validación.

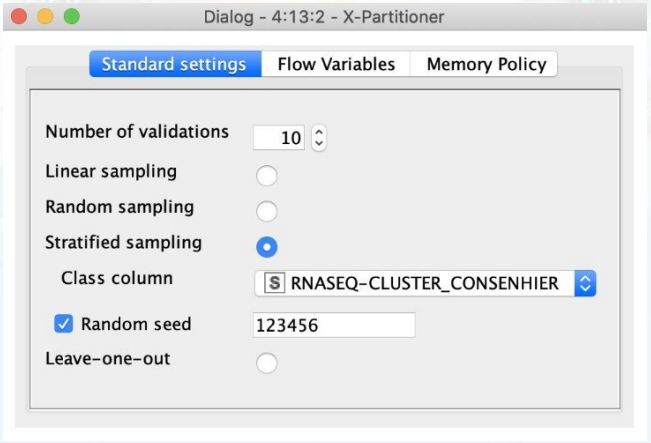


Figura 28: Opciones y parámetros del nodo X-Partitioner del Cross Validation

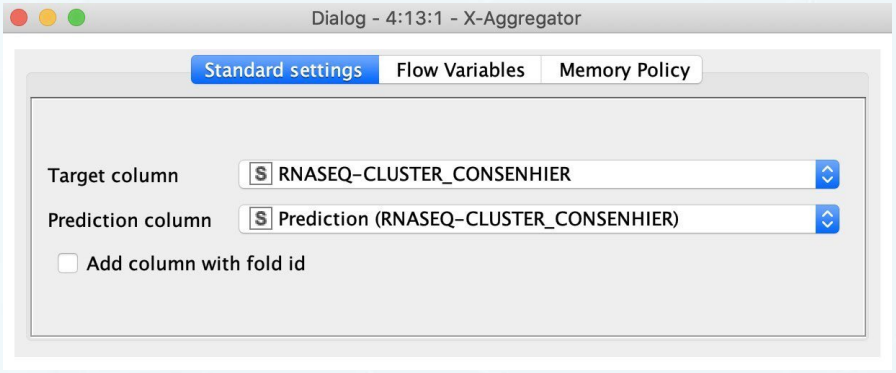


Figura 29: Opciones y parámetros del nodo X-Aggregator del Cross Validation

3.4 ¿CÓMO COMPARAR DISTINTOS ALGORITMOS?

En este apartado se va a detallar un flujo de datos que permita comparar distintos algoritmos sobre un mismo conjunto de datos. La comparación es similar a problemas de regresión y clasificación. Para no extender el material, vamos a realizar una validación cruzada con todos los algoritmos (KNN, árbol de decisión, SVM y Random Forest) y vamos a representarlos todos en una curva ROC (concepto detallado en el módulo 3). La Figura 30 muestra el flujo de datos para este ejemplo.

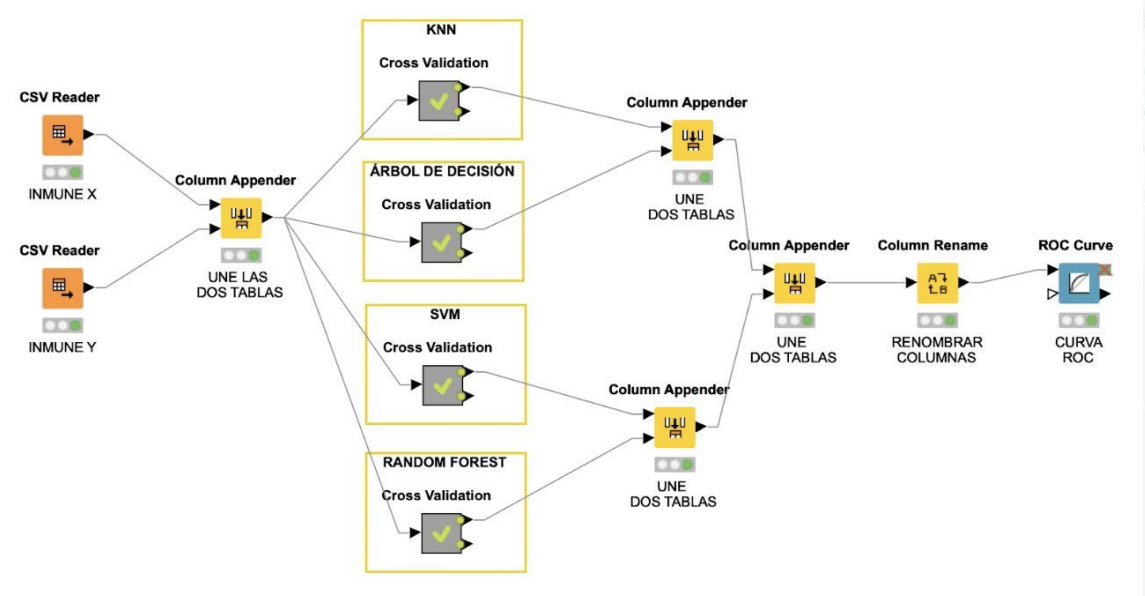


Figura 30: Flujo de datos de comparación de algoritmos

Siguiendo la línea de los ejemplos anteriores, vamos a utilizar el conjunto de datos inmune. Se leen los datos, al igual que hemos hecho en otros ejemplos y se utilizar el metanodo *Cross Validation* para realizar una validación cruzada a cada algoritmo. La salida de estos nodos, correspondiente a la tabla de predicción se va a unir con el nodo *Column Appender* para tener todos los resultados de predicción de todos los algoritmos en una única tabla. Luego se va a renombrar la columna correspondiente a las probabilidades de la clase positiva (en este caso la clase inmune) con el nombre de cada algoritmo para que sea interpretable en la curva ROC (Figura 31). Finalmente, se usa el nodo ROC Curve para crear la curva ROC donde se pueden ver las distintas áreas bajo la curva de cada uno de los algoritmos en una misma gráfica (Figura 32).

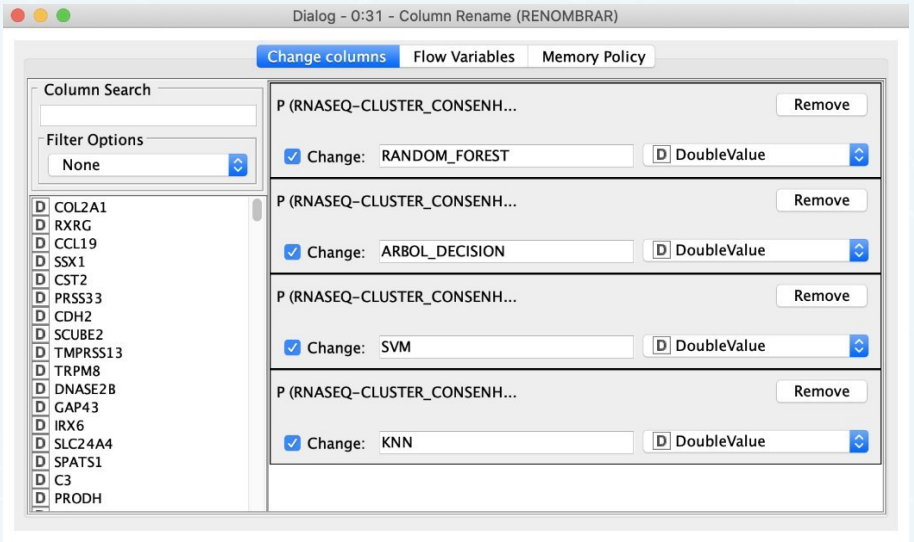


Figura 31: Opciones y parámetros del nodo Column Rename

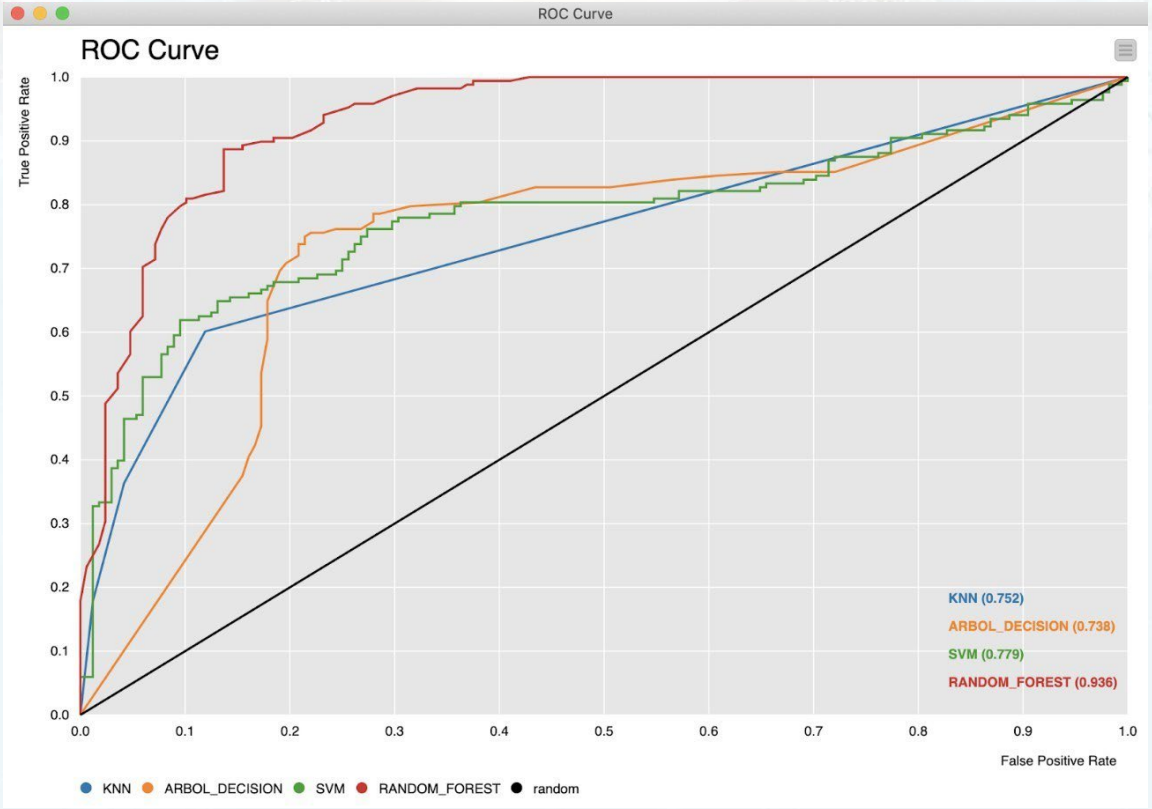


Figura 32: Flujo de datos de comparación de algoritmos – Curva ROC

4. REFERENCIAS BIBLIOGRÁFICAS

- Bakos, G. (2013). KNIME essentials. Packt Publishing Ltd.
- Blokdyk, G. (2019). KNIME a Complete Guide - 2019 Edition. Emereo Pty Limited, 2019.
- McCormick, K. (2019). Introduction to Machine Learning with KNIME. linkedin.com
- Silipo, R. (2016). Introduction to Data Analytics with KNIME: A Data Science Approach to Analytics. O'Reilly.
- Silipo, R., & Mazanetz, M. P. (2012). The KNIME cookbook. KNIME Press, Zürich, Switzerland.
- Strickland, J. (2016). Data Analytics Using Open-Source Tools. Lulu. com.