

Modulo IV.1

Técnicas de Observación y Evaluación a partir de Recursos Inteligentes: Introducción a la Minería de Datos



Co-funded by
the European Union



Dr. Álvaro Arnaiz González
Dr. José Francisco Díez Pastor
Dra. Sandra Rodríguez Arribas

“ El proyecto “(nombre del proyecto)” está cofinanciado por el programa Erasmus+ de la Unión Europea. El contenido de (esta nota de prensa/comunicado/publicación/etc.) es responsabilidad exclusiva del (nombre del centro educativo u organización de educación y formación) y ni la Comisión Europea, ni el Servicio Español para la Internacionalización de la Educación (SEPIE) son responsables del uso que pueda hacerse de la información aquí difundida. ”



Técnicas de Observación y Evaluación a partir de Recursos Inteligentes: Introducción a la Minería de Datos

1. Minería de Datos
2. Tipos de aprendizaje en minería de datos
3. Algoritmos de Clasificación
4. Algoritmos de *Clustering*
5. Algoritmos de Regresión
6. Knime
7. Material Adicional: Usando Knime

1. MINERÍA DE DATOS

La **minería de datos** también conocida como **Data Mining (DM)** en inglés, es el proceso de búsqueda y análisis en grandes bases de datos para encontrar información útil que sirva para la toma de decisiones.

Existen numerosas técnicas de **DM** que emplean el análisis matemático para deducir los patrones y tendencias que existen en los datos. Normalmente, estos patrones no se pueden detectar mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque el volumen de datos que hay que analizar es demasiado grande.

En la actualidad en campo de la **minería de datos** se emplea continuamente para el análisis de grandes cantidades de datos en diversos campos de conocimiento como la educación, la economía, los negocios, el medio ambiente...



1.1 CONCEPTOS BÁSICOS EN MINERÍA DE DATOS

Conjunto de datos o *Data set*

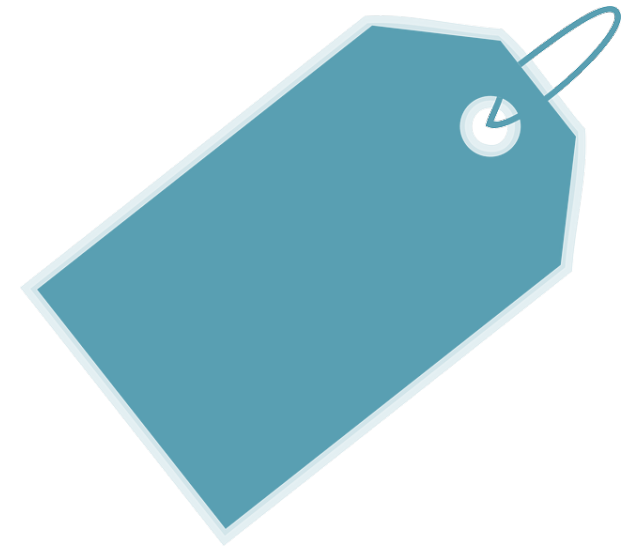
Es una colección grande de datos generalmente organizados en filas y columnas que contienen variables y atributos. Cada uno de estos valores se conoce con el nombre de dato. El conjunto de datos también puede consistir en una colección de documentos o de archivos.



1.1 CONCEPTOS BÁSICOS EN MINERÍA DE DATOS

Clases o etiquetas

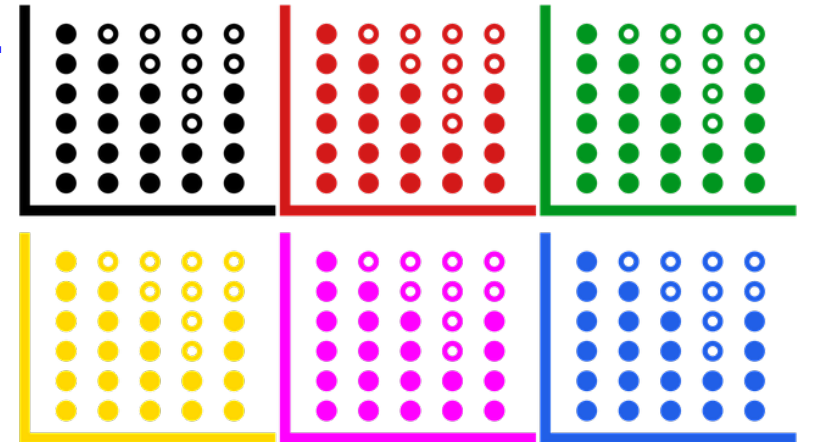
En el campo de la minería de datos, una clase es el atributo discreto cuyo valor se desea predecir en función de los valores de otros atributos. También se conoce como etiqueta.



1.1 CONCEPTOS BÁSICOS EN MINERÍA DE DATOS

Instancia

Una instancia es cada uno de los datos de los que se disponen para hacer un análisis. Cada instancia, a su vez, está compuesta de características que la describen. Por ejemplo, en una hoja de cálculo, las instancias serían las filas y las características la información almacenada en las columnas.



1.1 CONCEPTOS BÁSICOS EN MINERÍA DE DATOS

Algoritmo

En informática un algoritmo es un conjunto de instrucciones definidas, ordenadas y acotadas para resolver un problema, realizar un cálculo o desarrollar una tarea. En otras palabras, es un procedimiento paso a paso para obtener un resultado.



1.2 PROCESO DE APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS

Definición del problema

Preparación y recopilación de datos

Modelado y evaluación

Despliegue

1.2 PROCESO DE APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS

Definición del problema

- Es la primera fase en la que se traduce un problema específico en un problema de **minería de datos** en el que se plantean los objetivos del análisis y las preguntas de investigación.

1.2 PROCESO DE APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS

Preparación y recopilación de datos

- Es la fase más extensa del proceso ya que la calidad de los datos es uno de los retos más importantes en la **minería de datos**. Los datos brutos deben ser identificados, limpiados y almacenados en un formato preestablecido.

1.2 PROCESO DE APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS

Modelado y evaluación

- En este paso se seleccionan y aplican diferentes técnicas de modelado de datos (algoritmos) y después se establecen los parámetros y valores óptimos de dichas técnicas.

1.2 PROCESO DE APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS

Despliegue

- Es la última fase en la que se organizan y presentan los resultados de la **minería de datos** mediante gráficos e informes.

1.2 PROCESO DE APLICACIÓN DE TÉCNICAS DE MINERÍA DE DATOS

Es importante señalar que todo proceso de **minería de datos** es un proceso iterativo, lo que significa que el proceso no se detiene cuando se despliega una solución concreta. Puede ser sólo una nueva entrada para otro proceso de **minería de datos** (Rodríguez-Arribas, 2021). Es decir, en numerosas ocasiones la aplicación de técnicas de **DM** requiere de varias iteraciones y del empleo de algoritmos diferentes para poder extraer los resultados finales de la investigación que estamos realizando.



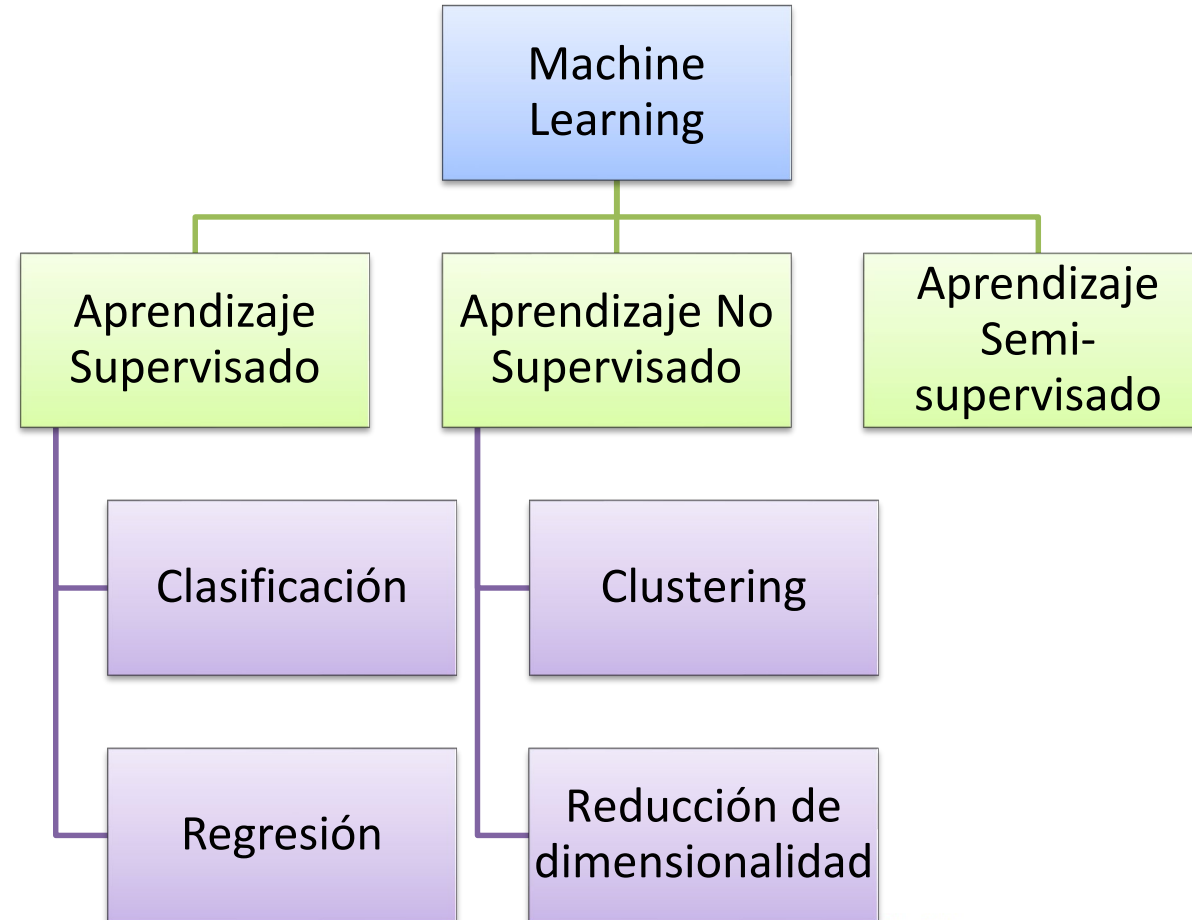
2. TIPOS DE APRENDIZAJE EN MINERÍA DE DATOS

Existen numerosas clasificaciones de los algoritmos que se emplean en el mundo de la **Minería de Datos**, pero es fundamental entender que hay dos enfoques básicos: el aprendizaje supervisado y el aprendizaje no supervisado.

Cuando tenemos que decidir qué algoritmo se empleará para realizar el análisis de los datos es importante tener en cuenta que tipo de aprendizaje se está utilizando, es decir, si se está hablando de aprendizaje supervisado o no supervisado (García, Luengo y Herrera, 2015). De acuerdo con el tipo de aprendizaje utilizado se emplearán diferentes técnicas y algoritmos como puede observarse en el siguiente esquema.



2. TIPOS DE APRENDIZAJE EN MINERÍA DE DATOS



2.1 APRENDIZAJE SUPERVISADO

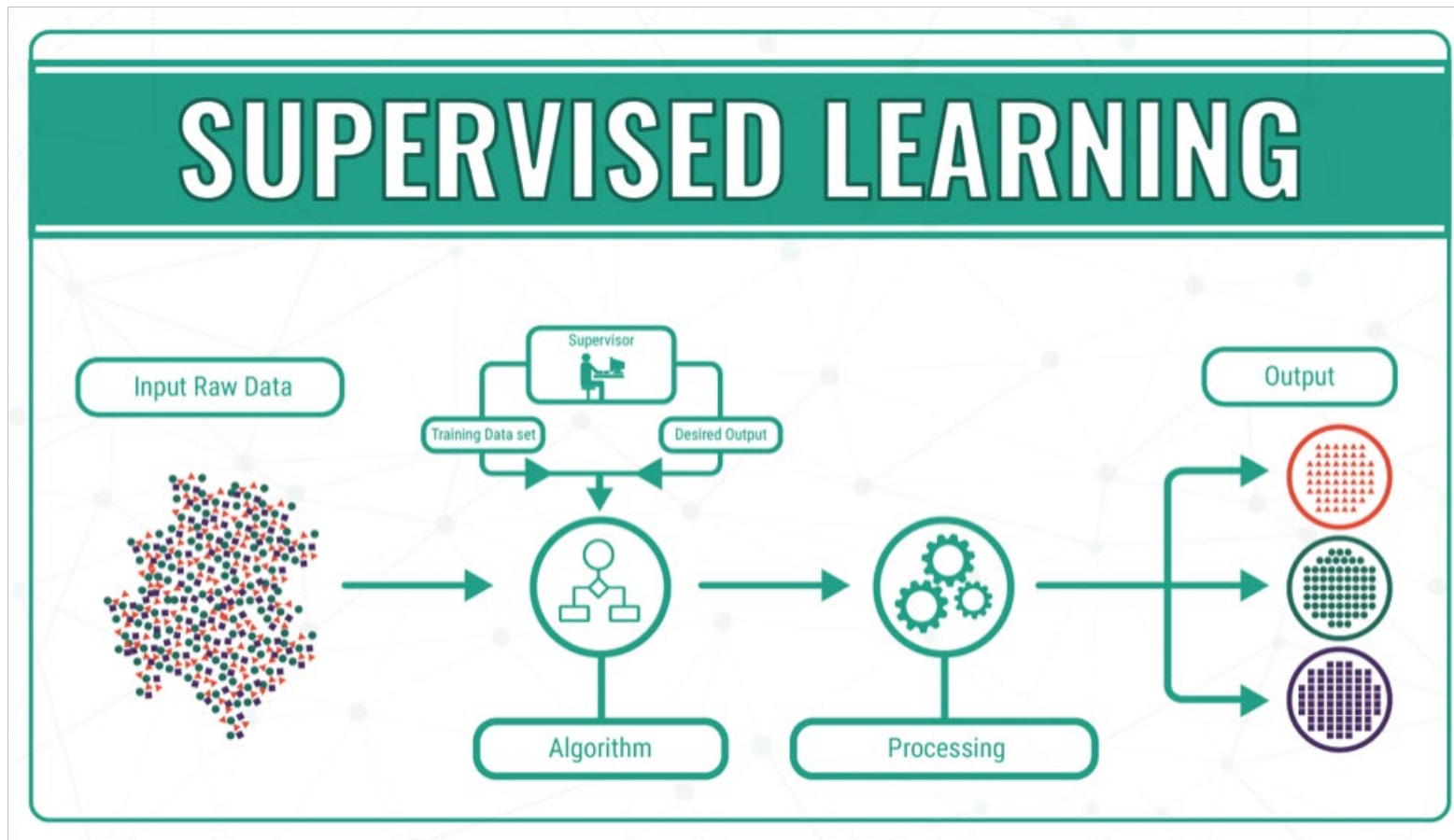
El objetivo fundamental del aprendizaje supervisado es la creación de un modelo que sea capaz de predecir valores correspondientes a objetos de entrada después de haberse familiarizado con una serie de ejemplos, los datos de entrenamiento.

Esta técnica consta de **dos pasos fundamentales**:

1. Una fase de entrenamiento donde se utiliza un conjunto de datos etiquetados, que contienen los datos de entrada y los resultados deseados para esos datos de entrenamiento con un algoritmo que permita deducir una función a partir de los datos que le estamos proporcionando al algoritmo
2. La fase de prueba, en donde se utiliza la función obtenida en el paso anterior para generar nuevas predicciones con nuevos conjuntos de datos.



2.1 APRENDIZAJE SUPERVISADO



2.1 APRENDIZAJE SUPERVISADO

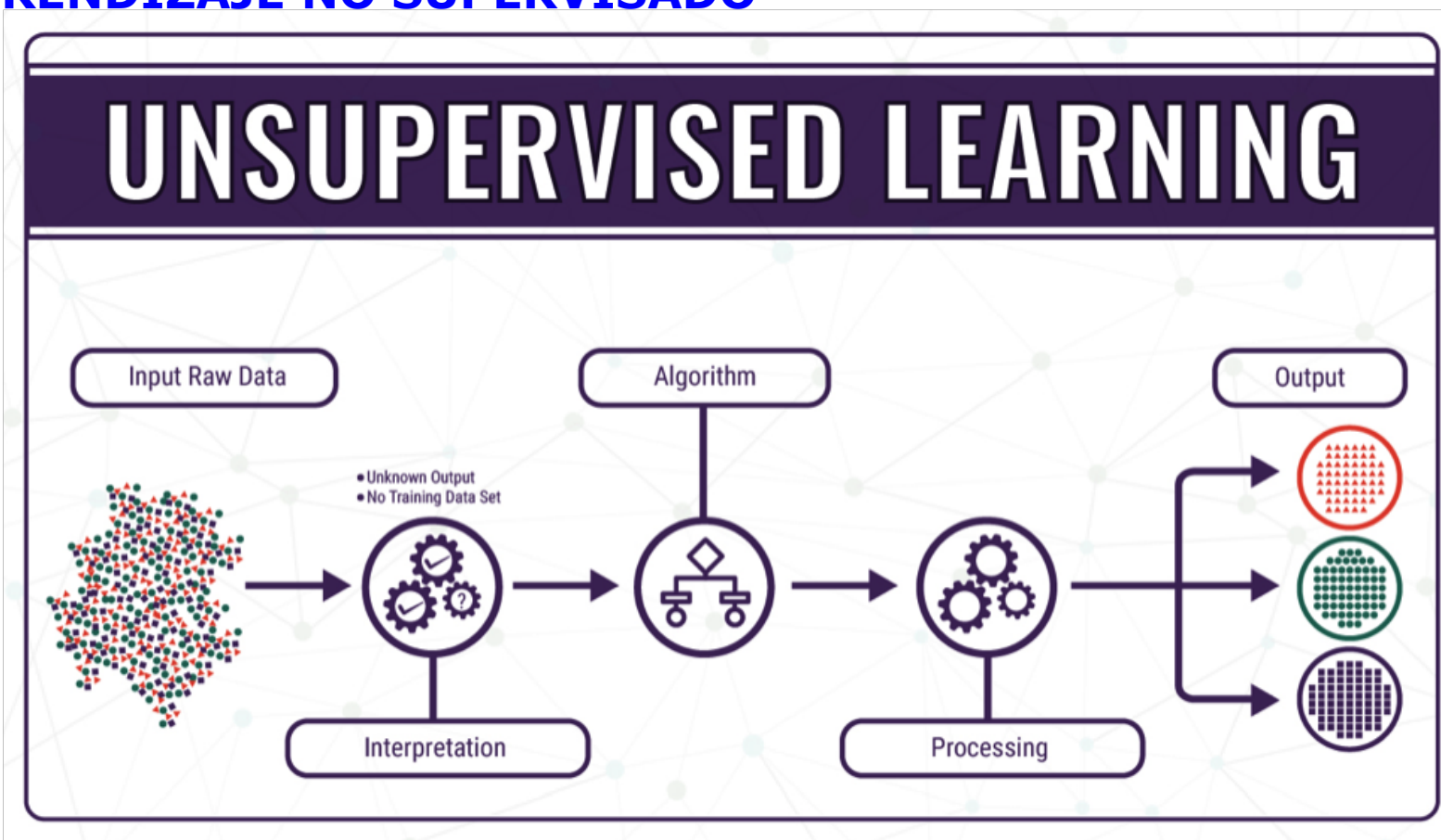
El proceso es conocido como aprendizaje supervisado, pues al conocer las respuestas de cada ejemplo del conjunto de entrenamiento, es posible corregir la función generada por el algoritmo. Se supervisa el entrenamiento del algoritmo mediante la corrección de parámetros del mismo, según sean los resultados que se obtienen de forma iterativa.

2.2 APRENDIZAJE NO SUPERVISADO

Este tipo de aprendizaje es el otro enfoque básico del **Machine Learning (ML)**. El aprendizaje no supervisado tiene datos sin etiquetar que el algoritmo tiene que intentar entender por sí mismo.

El objetivo de este tipo de aprendizaje es dejar que la máquina aprenda sin ayuda o indicaciones de los científicos de datos, es decir, sin supervisión y sin un conjunto de datos de entrenamiento. Además, la propia máquina realizará ajustes en los resultados y agrupaciones cuando haya resultados más adecuados, permitiendo que la máquina comprenda los datos y los procese de la mejor manera.

2.2 APRENDIZAJE NO SUPERVISADO



2.2 APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado se utiliza para explorar datos desconocidos y sin etiquetar. Puede revelar patrones que podrían haberse pasado por alto o examinar grandes conjuntos de datos que serían demasiado para que los abordara una sola persona.



2.3 APRENDIZAJE SEMI-SUPERVISADO

Actualmente se están realizando numerosas investigaciones con métodos de aprendizaje semi-supervisado. Estas técnicas de aprendizaje automático utilizan datos de entrenamiento tanto etiquetados como no etiquetados: normalmente una pequeña cantidad de datos etiquetados junto a una gran cantidad de datos no etiquetados (Zhu y Goldberg, 2009). Es decir, buscan mejorar los modelos de predicciones que se obtienen al utilizar exclusivamente datos etiquetados explorando la información estructural que contienen los datos no etiquetados.

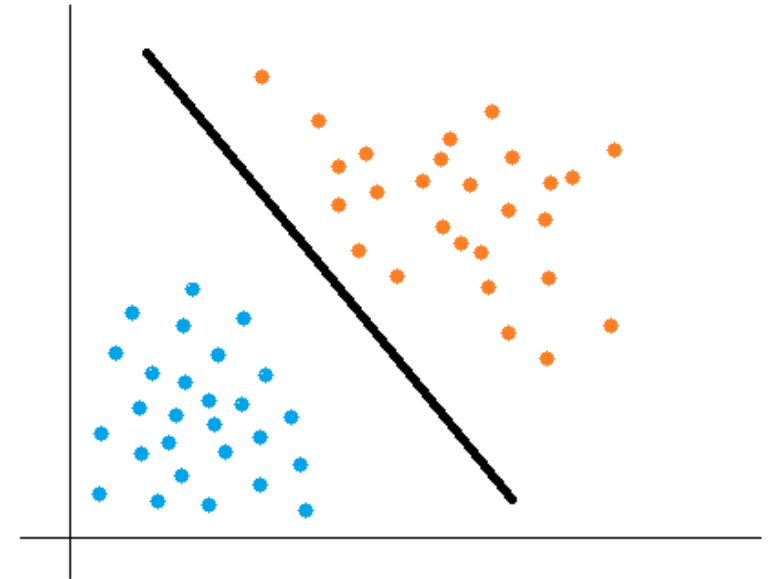
Podemos decir el aprendizaje semi-supervisado trata de combinar los dos enfoques tradicionales de la **minería de datos** (aprendizaje supervisado y aprendizaje no supervisado) para quedarse con lo mejor de cada uno de ellos.



3. ALGORITMOS DE CLASIFICACIÓN

Los algoritmos de clasificación son aquellos que utilizamos cuando el resultado esperado es una etiqueta discreta. Es decir, son útiles cuando la respuesta a la pregunta de investigación se encuentra dentro de un conjunto finito de resultados posibles.

La clasificación es muy similar al proceso de aprendizaje de las personas, ya que poseemos la capacidad de clasificar alimentos, libros, animales, planetas, es decir, todo lo que nos rodea.



3. ALGORITMOS DE CLASIFICACIÓN

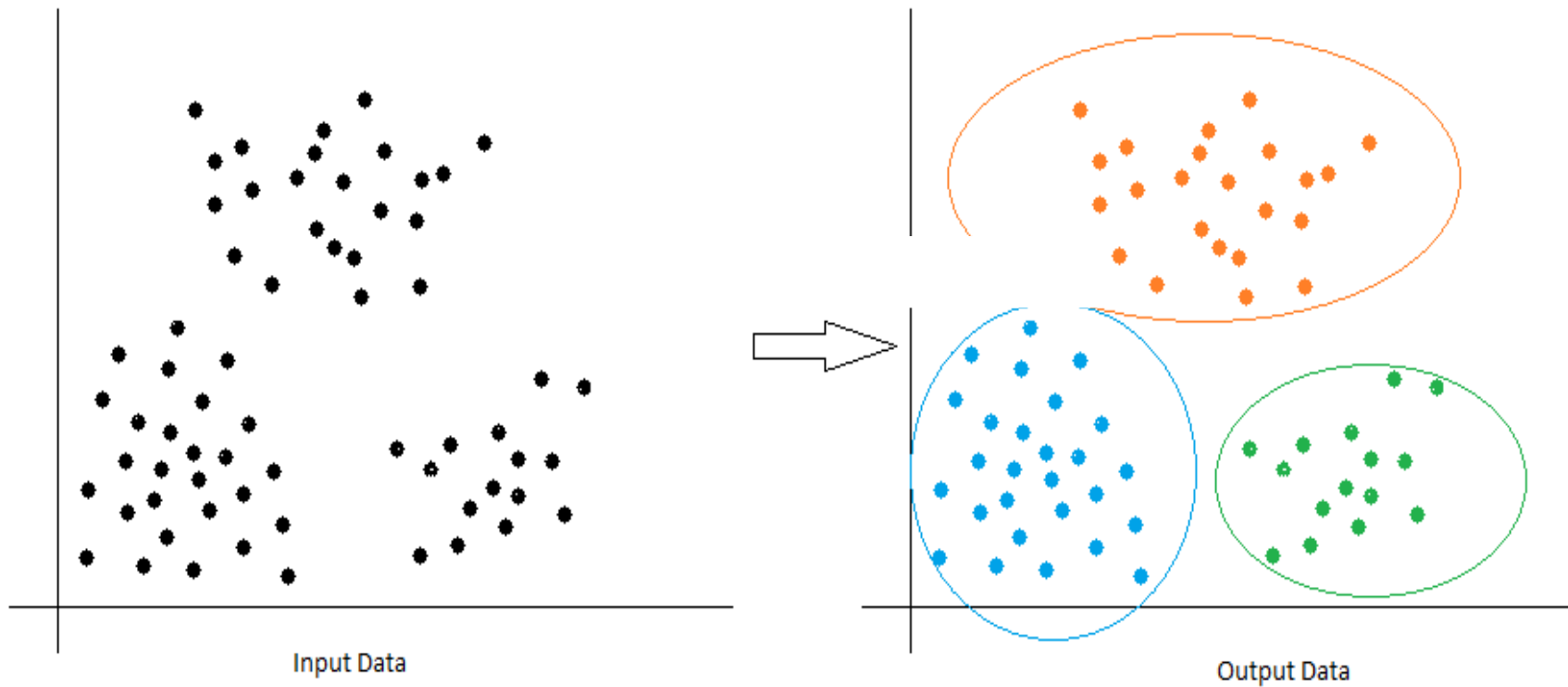
Estos algoritmos trabajan generalmente sobre la información entregada por un conjunto de muestras, patrones, ejemplos o prototipos de entrenamiento que son tomados como representantes de las clases, y los mismos conservan una etiqueta de clase correcta. A este conjunto de prototipos correctamente etiquetados se les llama conjunto de entrenamiento, y es el conocimiento disponible para la clasificación de nuevas muestras. El objetivo de la clasificación supervisada es determinar, según lo que se tenga conocimiento, cual es la clase a la que debería concernir una nueva muestra, teniendo en cuenta la información que se pueda extraer.

4. ALGORITMOS DE *CLUSTERING*

Los algoritmos de agrupamiento o de *clustering* se encargan de agrupar los objetos de un conjunto de datos en función de sus similitudes. De este modo los objetos que están dentro de un clúster o grupo tienen más similitudes entre ellos que diferencias.

Estos algoritmos trabajan con datos no etiquetados por lo que es el propio algoritmo el que analiza los datos para encontrar el número de agrupamientos óptimo para el conjunto de datos de entrada ya que no disponemos de conocimientos previos sobre las características de los datos y sus clases.

4. ALGORITMOS DE CLUSTERING



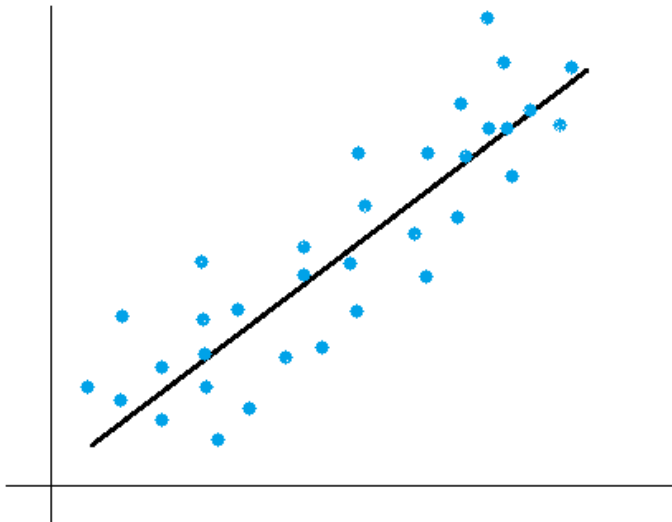
4. ALGORITMOS DE *CLUSTERING*

Los agrupamientos que realizan los algoritmos pueden ser de dos tipos:

1. Clúster duro: cada dato pertenece exclusivamente a un grupo

2. Clúster blando (difuso): los datos pueden pertenecer a varios grupos en distintos grados, es decir, un mismo dato puede tener un grado de pertenencia del 60% al grupo 1 y del 40% en el grupo 2.

5. ALGORITMOS DE REGRESIÓN



Los algoritmos de regresión es un subcampo del aprendizaje supervisado cuyo objetivo es establecer un método para la relación entre un cierto número de características y una variable objetivo continua.

Se trata de algoritmos que establecen una recta para proporcionar la tendencia de un conjunto de datos, es decir, el fin de estos algoritmos es relacionar un número de características y una variable objetivo continua.

5. ALGORITMOS DE REGRESIÓN

Esta técnica es útil para predecir resultados que son valores continuos, eso significa que la respuesta a la pregunta de investigación se presenta mediante una cantidad que puede determinarse de manera flexible en función de las entradas del modelo en lugar de limitarse a un conjunto de etiquetas finito como en el caso de la clasificación.



6. KNIME

6. 1 INTRODUCCIÓN

KNIME es una aplicación de código abierto (*open source*) que permite aplicar a nuestros propios conjuntos de datos o a conjuntos de datos de ejemplo:

- Métodos estadísticos
- Algoritmos de **minería de datos** o aprendizaje automático.
- Técnicas de visualización.

Está construido sobre la plataforma Eclipse y está programado en Java. Al tratarse de un software de código abierto tiene muchas ventajas, su código pertenece a la comunidad de usuarios y desarrolladores, lo que garantiza que siempre será una herramienta libre y gratuita que puede descargarse y utilizarse sin tener que pagar bajo los términos de licencia GPLv3. Además, permite la incorporación de Código en R o Python.



6. KNIME

Es una herramienta de "Programación visual". El análisis de los datos se puede realizar de forma intuitiva configurando el proceso simplemente haciendo clics con el ratón. Se colocan los "nodos" que necesitamos, sin necesidad de conocer su nombre o como se configuran, puesto que en todo momento disponemos de ayudas

Se trata de una herramienta diseñada para ser sencilla de usar. El concepto más importante en el uso de la herramienta es el de *workflow* (en castellano, flujo de trabajo). Un flujo de trabajo es una secuencia de pasos configurada por el usuario. Formalmente es un conjunto de nodos unidos entre sí con flechas que representan el flujo de los datos entre los nodos. Un nodo encapsula distintos trabajos que se pueden realizar con los datos, existen nodos para muchas tareas



6. KNIME

Existen nodos para:

- a. Cargar datos desde ficheros o bases de datos.
- b. Crear, modificar o eliminar filas o columnas del conjunto de datos con el que estemos trabajando.
- c. Calcular estadísticas: medias, percentiles, correlaciones etc.
- d. Combinar datos de fuentes de datos distintas.
- e. Construir y evaluar modelos de aprendizaje automático como: clasificación, regresión o clustering.
- f. Visualizar los datos usando gráficos de barras, tarta, dispersión y también otros tipos de gráficos más avanzados.
- g. Generación de informes.

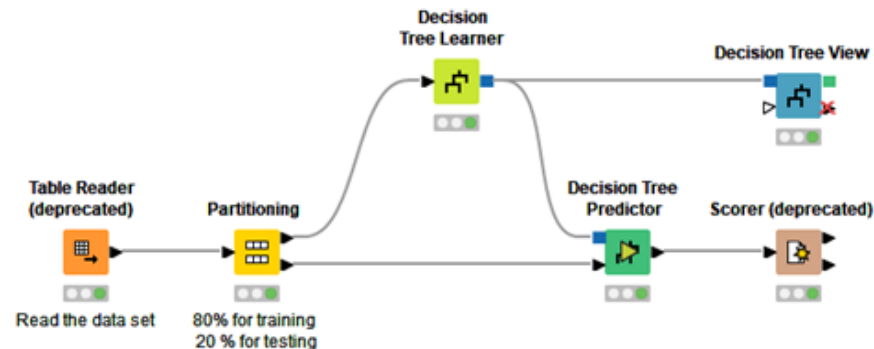


6. KNIME

Un flujo de trabajo podría tener un nodo para cargar un conjunto de datos a partir de un fichero de Excel, a continuación, un nodo para seleccionar atributos (columnas) de dicho conjunto de datos y por último otro nodo para visualizar estadísticas de los atributos seleccionados.

Decision Tree: Training

This workflow is an example of how to build a basic prediction / classification model using a decision tree.
Dataset describes wine chemical features. Output class is wine color: red/white



Bibliografía

- García, S., Luengo, J., y Herrera, F. (2015). Data Preprocessing in Data Mining / by Salvador García, Julián Luengo, Francisco Herrera. Springer
- Sáiz-Manzanares, M.C., Marticorena, R., y Arnaiz-Gonzalez, Á. (2022). Improvements for therapeutic intervention from the use of web applications and machine learning techniques in different affectations in children aged 0-6 years. *Int. J. Environ. Res. Public Health*, 19, 6558. <https://doi.org/10.3390/ijerph19116558>
- Sáiz-Manzanares, M.C., Marticorena, R., & Arnaiz, Á. (2020). Evaluation of Functional Abilities in 0–6-Year-Olds: An Analysis with the eEarlyCare Computer Application. (2020). *Int. J. Environ. Res. Public Health*, 17(9), 3315, 1-17 <https://doi.org/10.3390/ijerph17093315>
- Sáiz-Manzanares, M.C., Marticorena, R., Arnaiz-González, Á., Díez-Pastor, J.F., & Rodríguez-Arribas, S. (2019, March). Computer application for the registration and automation of the correction of a functional skills detection scale in Early Care. 13th International Technology, Education and Development Conference Proceedings of INTED2019 Conference 11th-13th (5322-5328). IATED: Valencia. doi: 10.21125/inted.2019.1320

Imágenes

Imagen 1 <https://pixabay.com/es/illustrations/grande-datos-teclado-computadora-895567/>

Imagen 2 <https://pixabay.com/es/vectors/etiqueta-equipaje-blanco-precio-309129/>

Imagen 3 <https://pixabay.com/es/illustrations/es-el-gr%c3%a1fico-tabla-varilla-5474235/>

Imagen 4 <https://pixabay.com/es/vectors/aprendizaje-autom%c3%a1tico-7271039/>

Imagen 5 <https://pixabay.com/es/photos/pir%c3%a1mide-gr%c3%a1fico-colores-infografia-2611048/>

Imagen 6 <https://chisoftware.medium.com/supervised-vs-unsupervised-machine-learning-7f26118d5ee6>

Imagen 7 <https://chisoftware.medium.com/supervised-vs-unsupervised-machine-learning-7f26118d5ee6>

Imagen 8 elaboración propia

Imagen 9 elaboración propia

Imagen 10 elaboración propia

Imagen 11 elaboración propia

Modulo IV.1

Material Adicional: Usando KNIME



Co-funded by
the European Union



e-EarlyCare-T



Material Adicional: Usando KNIME

1. **INSTALACIÓN DE KNIME.**
2. **EL *WORKFLOW* DE KNIME**
 1. **Nodos**
 2. **El área de trabajo.**
3. **Ejemplo genérico: Clasificando especies de flores.**
4. **Ejemplo con datos de intervención terapéutica inteligente (EarlyCare)**

1. INSTALACIÓN DE KNIME

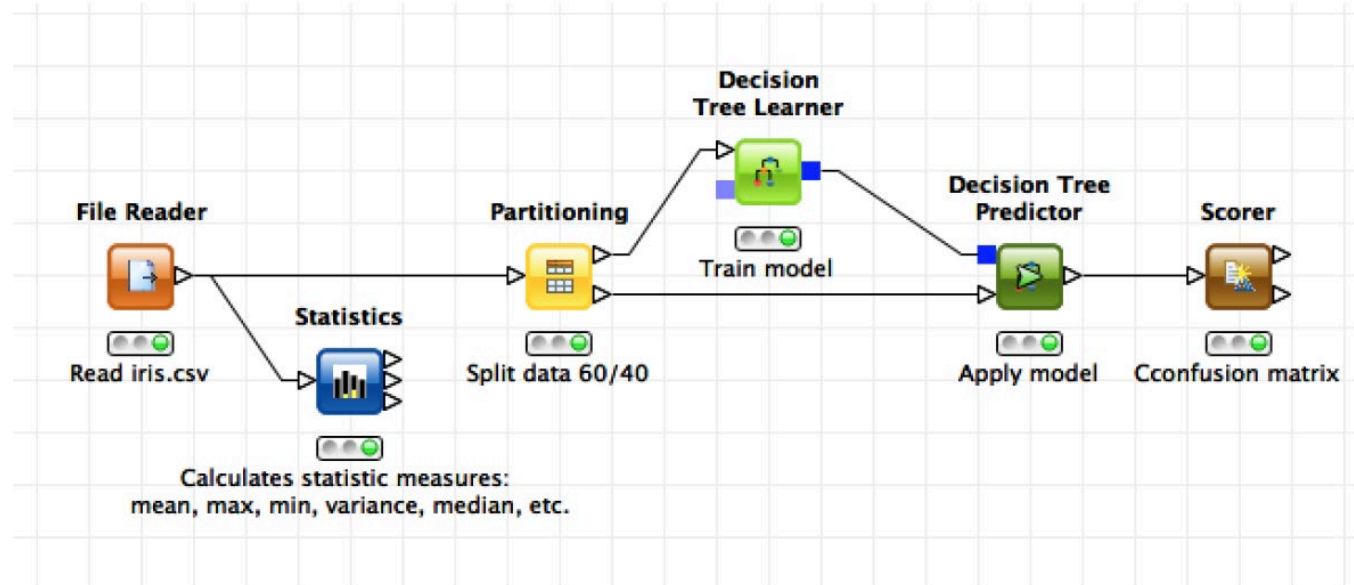
KNIME es una aplicación Java, lo que significa que será necesario tener instalada la máquina virtual de Java antes de poder instalar y ejecutar el programa.

Para instalar el software deberemos ir a <https://www.knime.com/downloads> , una vez allí descargaremos “KNIME Analytics Platform” eligiendo la versión que corresponda para el ordenador personal del que dispongamos: Mac, Windows 32 bits (ordenadores antiguos), Windows 64 (ordenadores modernos) o Linux.



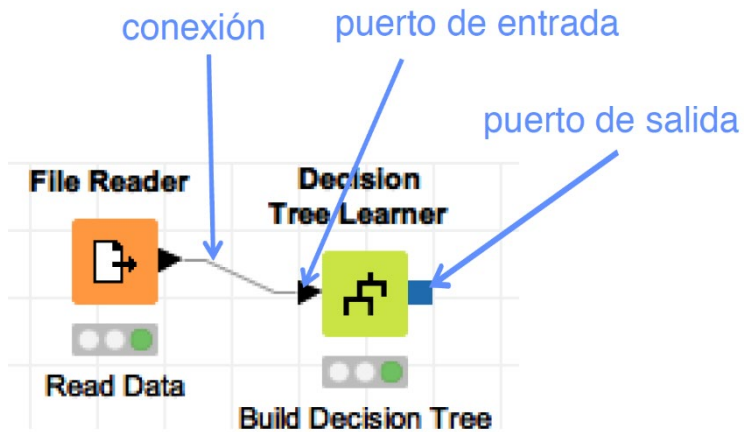
2. EL WORKFLOW DE KNIME

Son una representación visual de la secuencia de pasos que tiene lugar en el proceso de análisis de datos. Están compuestos de una serie de nodos enlazados.



2.1 NODOS

Los nodos encapsulan los algoritmos que implementan las acciones que pueden ejecutarse sobre los datos:



- Manipulación de filas, columnas, etc
- Creación de modelos de minería de datos.
- Evaluación de modelos.
- Aplicación de modelos sobre nuevos datos.
- Procesos ETL (Extract, Load, Transform).
- Creación de informes a medida.

2.2 EL ÁREA DE TRABAJO

The screenshot displays the KNIME Analytics Platform interface. The main workspace shows a workflow with three nodes: **Partitioning** (Random drawing 80% upper port 20% lower port), **Decision Tree Learner** (Train to predict class "income"), and **Decision Tree Predictor** (Apply decision tree model to test set). The **Decision Tree Learner** node is selected, and its description is shown in the right-hand pane. The description states: "This node induces a classification decision tree in main memory. The target attribute must be nominal. The other attributes used for decision making can be either nominal or numerical. Numeric splits are always binary (two outcomes), dividing the domain in two partitions at a given split point. Nominal splits can be either binary (two outcomes) or they can have as many outcomes as nominal values. In the case of a binary split the nominal values are divided into two subsets. The algorithm provides two quality measures for split calculation; the gini index and the gain ratio. Further, there exist a post pruning method to reduce the tree size and increase prediction accuracy. The pruning method is".

The **Workflow Coach** pane on the left lists recommended nodes:

Node	Community
Decision Tree Predictor	85%
Decision Tree To Image	5%
Decision Tree to Ruleset	3%
PMML Writer	3%
Decision Tree View	1%
PMML To Cell	<1%
Boosting Learner Loop End	<1%
Model Writer	<1%
Model Loop End	<1%

The **Node Repository** pane on the left shows a tree view of nodes categorized by IO, Manipulation, Views, Analytics, DB, Other Data Types, Structured Data, Scripting, Tools & Services, Workflow Control, Workflow Abstraction, and Reporting.

The **Console** pane at the bottom right shows the following output:

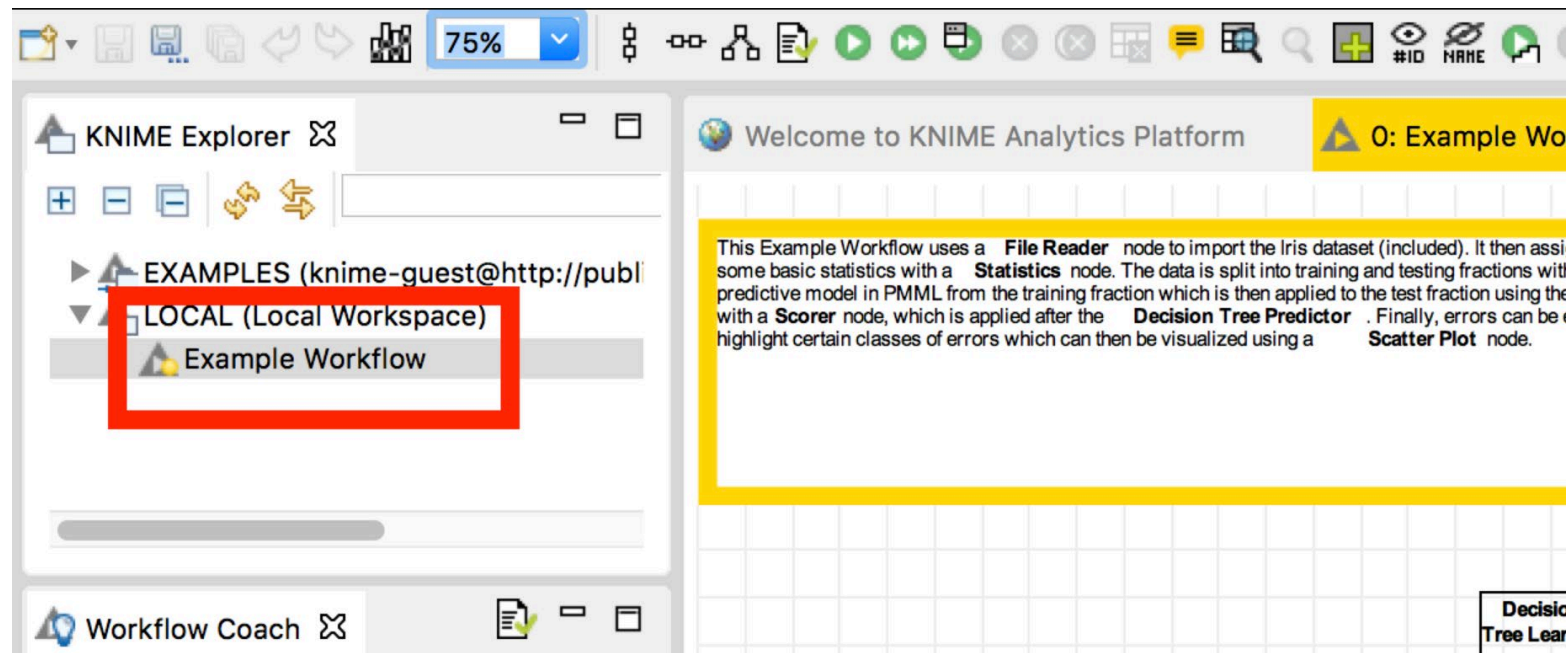
```
*** Welcome to KNIME Analytics Platform v4.0.1.v201908131317 ***
*** Copyright by KNIME AG, Zurich, Switzerland ***
Log file is located at: /Users/cgosorio/knime-workspace/.metadata/knime/kni
WARN Color Manager 0:2 Column "income" has no nominal values
```

2.2 EL ÁREA DE TRABAJO

El *Workspace* (espacio de trabajo), es la carpeta o directorio de nuestro ordenador donde están almacenados todos los proyectos realizados con KNIME. Será necesario elegir un espacio de trabajo antes de arrancar el programa (también se puede dejar la carpeta que aparece por defecto al realizar la instalación).

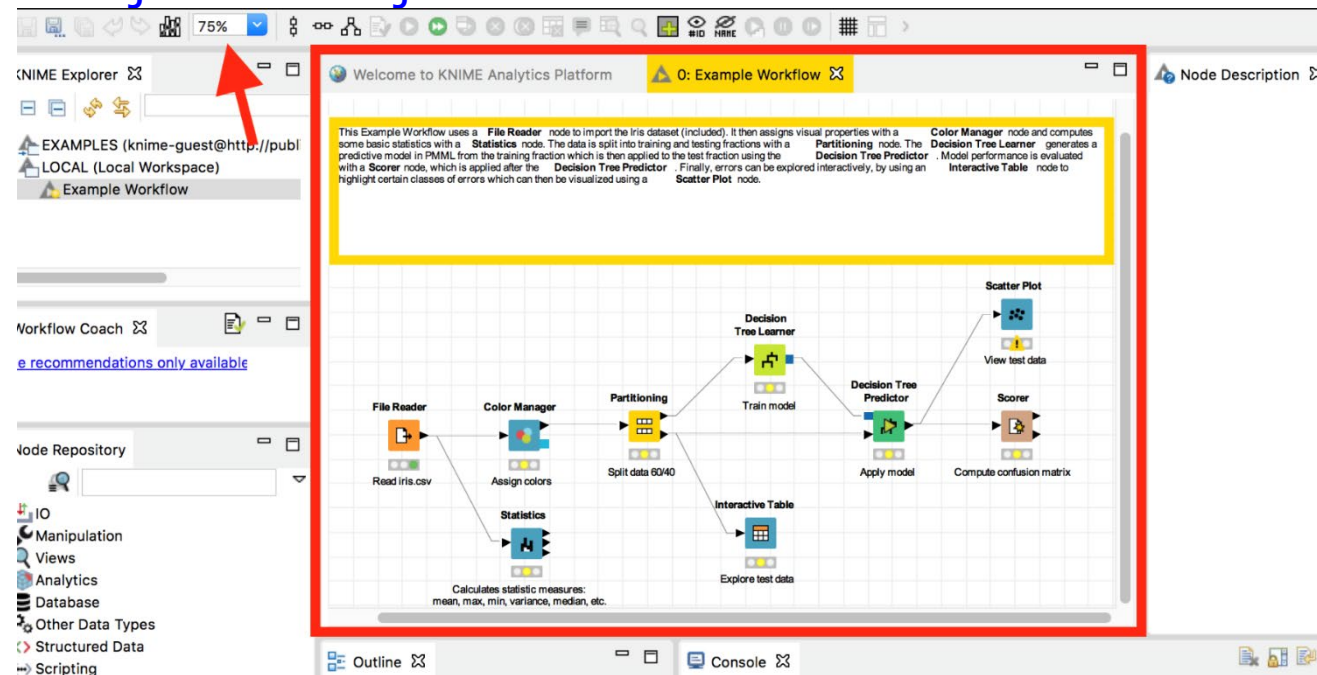
2.2 EL ÁREA DE TRABAJO: KNIME Explorer

Es el área donde se gestionan los proyectos y flujo de trabajo guardados. Donde se importan o exportan los flujos de trabajo



2.2 EL ÁREA DE TRABAJO: *Workflow editor.*

Es la zona de trabajo principal, donde se arrastran los nodos, se conectan entre si y se configura el flujo de trabajo.



2.2 EL ÁREA DE TRABAJO: *Outline*.

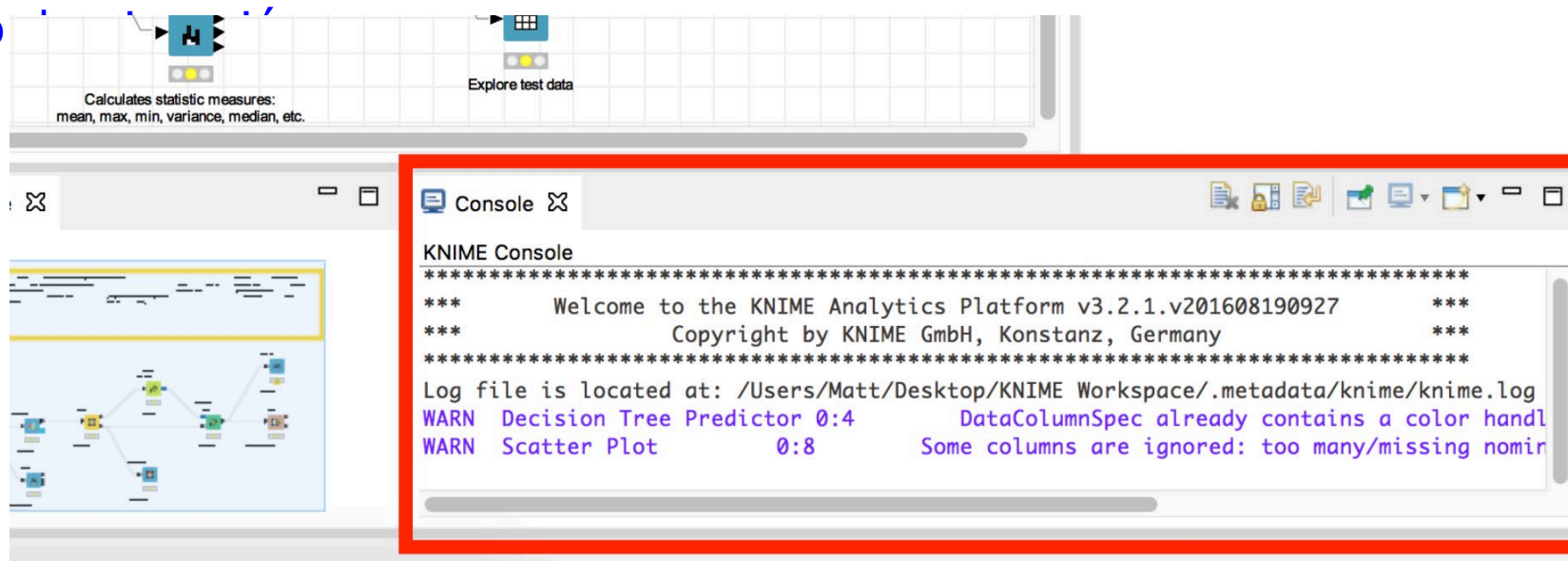
Muestra una visión global del flujo de trabajo, para facilitar el movernos de una parte a otra cuando el flujo de trabajo es muy grande.

The screenshot displays the KNIME software interface. On the left, a vertical sidebar lists various tool categories: IO, Manipulation, Views, Analytics, Database, Other Data Types, Structured Data, Scripting, Tool Integration, Community Nodes, KNIME Labs, Workflow Control, Social Media, Reporting, Chemistry, and ChemAxon / Infocom. The main workspace is a grid where two nodes are visible: 'Statistics' (with a description: 'Calculates statistic measures: mean, max, min, variance, median, etc.') and 'Interactive Table' (with a description: 'Explore test data'). A red rectangular box highlights the 'Outline' panel, which provides a hierarchical overview of the entire workflow. To the right of the Outline panel is the 'Console' window, which displays the following text:

```
KNIME Console
*****
***      Welcome to the KNIME Anc
***      Copyright by KN
*****
Log file is located at: /Users/Mat
WARN Decision Tree Predictor 0:4
WARN Scatter Plot           0:8
```

2.2 EL ÁREA DE TRABAJO: *Console*.

Es un campo de salida de texto que muestra avisos y errores que se producen al ejecutar el flujo de trabajo. También muestra información relevante sobre el proceso



2.2 EL ÁREA DE TRABAJO: *Node repository*.

Es la zona donde se encuentran los nodos organizados por categorías. También es posible buscar nodos por su nombre. Para usar un nodo simplemente se selecciona y se arrastra al editor.

The screenshot displays the KNIME Analytics Platform interface. On the left, the 'Node Repository' panel is highlighted with a red border, showing a tree view of nodes categorized by IO, Manipulation, Views, Analytics, Database, Other Data Types, Structured Data, Scripting, Tool Integration, Community Nodes, KNIME Labs, Workflow Control, Social Media, Reporting, Chemistry, and ChemAxon / Infocom. The main workspace shows a workflow with nodes: File Reader (Read Iris.csv), Color Manager (Assign colors), Partitioning (Split data 60/40), Decision Tree Learner (Train model), Decision Tree Predictor (Apply model), Scorer (Compute confusion matrix), Interactive Table (Explore test data), and Scatter Plot (View test data). The bottom right shows the 'Console' window with the following text:

```
KNIME Console
*****
*** Welcome to the KNIME Analytics Platform v3.2.1.v201608190927 ***
*** Copyright by KNIME GmbH, Konstanz, Germany ***
*****
Log file is located at: /Users/Matt/Desktop/KNIME Workspace/.metadata/knime.l
WARN Decision Tree Predictor 0:4 DataColumnSpec already contains a color hc
WARN Scatter Plot 0:8 Some columns are ignored: too many/missing nc
```

Material Adicional: Usando KNIME

2.2 EL ÁREA DE TRABAJO: *Workflow coach*.

Si hemos dado permiso para que recojan nuestros datos de uso, en este apartado se hacen sugerencias de cuáles son los nodos más probables que vamos a necesitar usar en cada momento.

Recommended Nodes	Community
Decision Tree Predictor	85%
Decision Tree To Image	5%
Decision Tree to Ruleset	3%
PMML Writer	3%
Decision Tree View	1%
PMML To Cell	<1%
Boosting Learner Loop End	<1%
Model Writer	<1%
Model Loop End	<1%

Try this:
KNIME's Interactive Visualizations:
1) Execute the workflow
2) Open the Scorer node view
3) Highlight a cell in the confusion matrix
4) Open the Interactive Table view
5) Select "Highlight" -> "Filter" -> "Show Highlighted Only"
This shows only the misclassified data rows.

2.2 EL ÁREA DE TRABAJO: *Node description.*

Es un cuadro informativo que aparece cuando se selecciona un nodo y muestra información sobre las tareas que realiza el nodo y sobre cuales son sus puertos (entradas y salidas).

The screenshot displays the KNIME Analytics Platform interface. The main workspace shows a workflow with the following nodes: File Reader (Read Iris.csv), Column Filter (Postal ONLY), Color Manager (Assign colors), Statistics (Calculates statistic measures: mean, max, min, variance, median, etc.), Partitioning (Split data 60/40), Decision Tree Learner (Train model), Decision Tree Predictor (Apply model), Interactive Table (Explore test data), Scorer (Compute confusion matrix), and Scatter Plot (View test data). A yellow box highlights a text description of the workflow. A red box highlights the 'Node Description' window for the 'File Reader' node, which contains the following text:

File Reader

Click on the table header

If the column header in the preview table is clicked, a new dialog opens where column properties can be set: name and type can be changed (and will be fixed then). A pattern can be entered that will cause a "missing cell" to be created when it's read for this column. Additionally, possible values of the column domain can be updated by selecting "Domain". And, you can choose to skip this column entirely, i.e. it will not be included in the output table then.

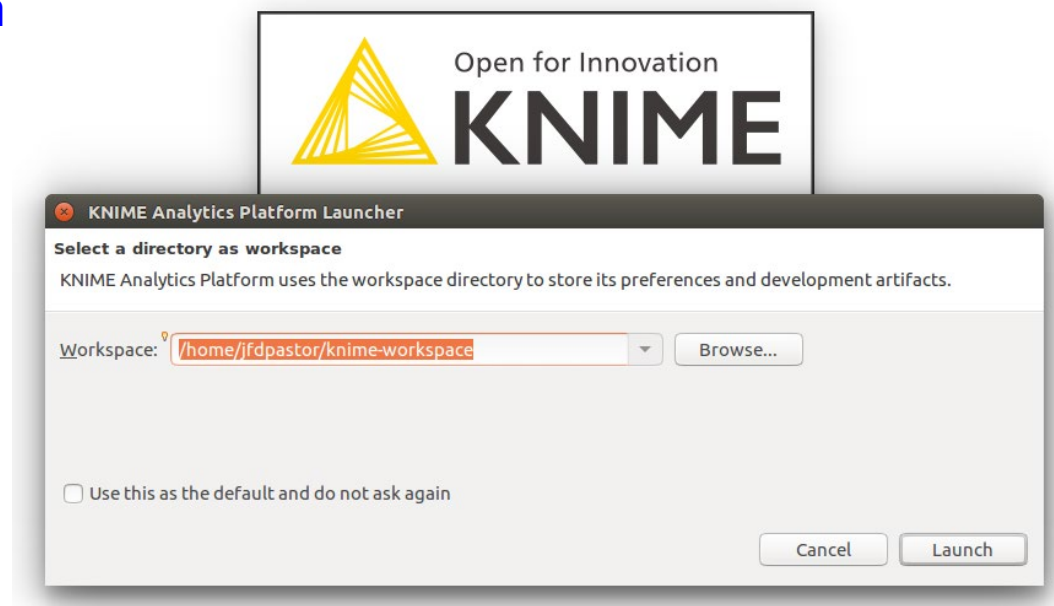
Ports

Output Ports

0 Datable just read from the file

3. Ejemplo genérico: Clasificando especies de flores.

Partiendo de que KNIME ya está instalado, vamos a la carpeta donde se encuentra y lo ejecutamos haciendo doble **click** en su icono. Al abrirse nos preguntará la carpeta del «Workspace». Esta es la proyectos.



3. Ejemplo genérico: Clasificando especies de flores.

The screenshot displays the KNIME Analytics Platform interface. The main window shows a "Welcome back" message and a search bar for workflows, nodes, and more. Below the message are three promotional cards: "Share your workflows and components on KNIME Hub", "KNIME Courses: learn all about Big Data, Text Mining and more", and "Questions? Ask the community".

On the left side, there are two panels:

- KNIME Explorer:** Shows a tree view of the workspace with folders for "My-KNIME-Hub (hub.knime.com)", "EXAMPLES (knime@hub.knime.com)", "LOCAL (Local Workspace)", and "Example Workflows".
- Workflow Coach:** Displays a table of recommended nodes from the community.

Recommended Nodes	Community
File Reader	24%
CSV Reader	18%
Excel Reader (XLS)	17%
Table Creator	12%
Database Reader (legacy)	7%
Table Reader	4%

Below the Workflow Coach is the **Node Repository**, which lists various node categories such as IO, Manipulation, Views, Analytics, DB, Other Data Types, Structured Data, Scripting, Tools & Services, Workflow Control, Workflow Abstraction, and Reporting.

At the bottom, there are two panels:

- Outline:** Shows a message: "An outline is not available."
- Console:** Displays the KNIME Console output, including the version information and the log file location: `/home/jfdpastor/knime-workspace/.metadata/knime/knime.log`.

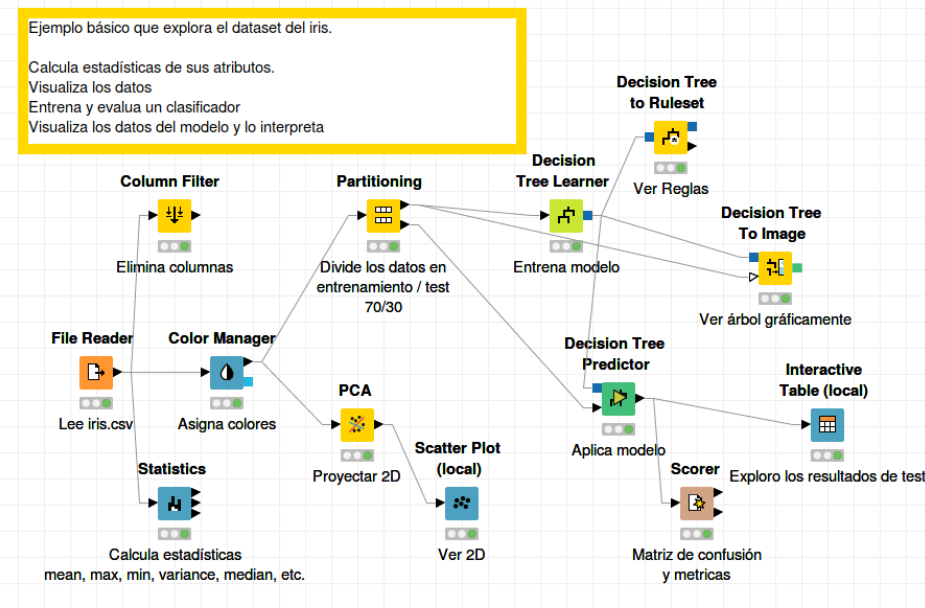
3. Ejemplo genérico: Clasificando especies de flores.

- Se necesita descargar el fichero zip llamado «Minería_Ejemplo.zip»
 - NO lo descomprimimos.
- En el Explorador de KNIME hacer click derecho y después «Import KNIME workflow ...»
- Posteriormente la opción «Select file» → «browse»
- Lo seleccionamos y damos «Ok»

3. Ejemplo genérico: Clasificando especies de flores.

Es un «workflow» básico, que trabaja con el conjunto de datos del iris. Iris está formado por 150 ejemplos pertenecientes a 3 especies de flores diferentes. Cada ejemplo tiene 4 atributos que describen a la flor: longitud del sépalo, longitud del pétalo, anchura del sépalo, anchura del pétalo. Con este conjunto de datos vamos a:

- Calcular estadísticas de sus atributos
- Visualizar los datos
- Entrenar y evaluar un clasificador.



3. Ejemplo genérico: Clasificando especies de flores.

En el Editor, podemos ver una serie de nodos interconectados. Sobre este editor se arrastran los nodos, se unen entre si, se configuran y se ejecutan para realizar operaciones y análisis sobre los datos.

Tiene herramientas de navegación como zoom *in/out* (hacer más grande o más pequeño) y permite añadir comentarios.

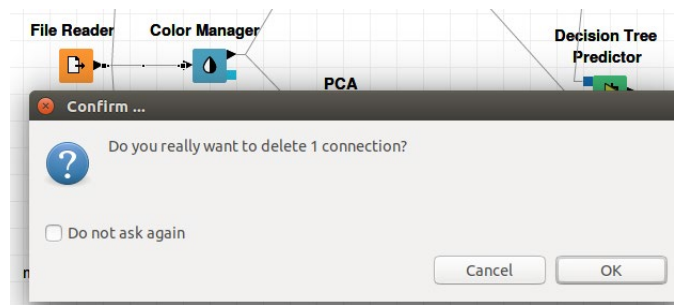
Podemos ejecutar cada nodo o todo el *workflow* completo con los botones similares a «*play*»



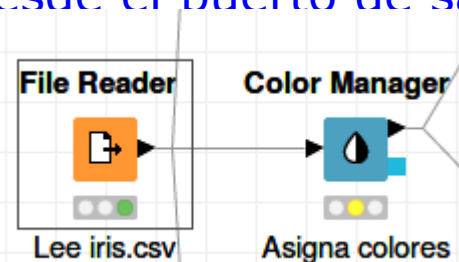
Es necesario ejecutar los nodos posteriores cada vez que se ejecuta un cambio en un nodo. Es decir, si cambiamos algún parámetro de un nodo que está al principio del flujo de trabajo, tenemos que dar al botón de *play* con dos flechas blancas para volver a ejecutar todos los nodos que están a continuación.

3. Ejemplo genérico: Clasificando especies de flores.

Vamos a practicar el borrado y creación de conexiones entre nodos. Podemos borrar por ejemplo la conexión entre «File Reader» y «Color Manager»

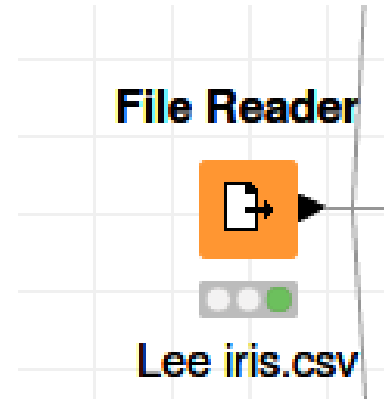


Para volver a crear la conexión: Seleccionamos ambos, botón derecho, «connect selected nodes» o bien arrastramos el ratón desde el puerto de salida de «File Reader» al puerto de entrada de «Color Manager»



3. Ejemplo genérico: Clasificando especies de flores. Cargando datos

El nodo «File Reader» es el nodo usado para cargar conjuntos de datos (leer los datos desde donde estén guardados). Puede cargar datos desde una url (internet) o desde el disco duro de nuestro ordenador.

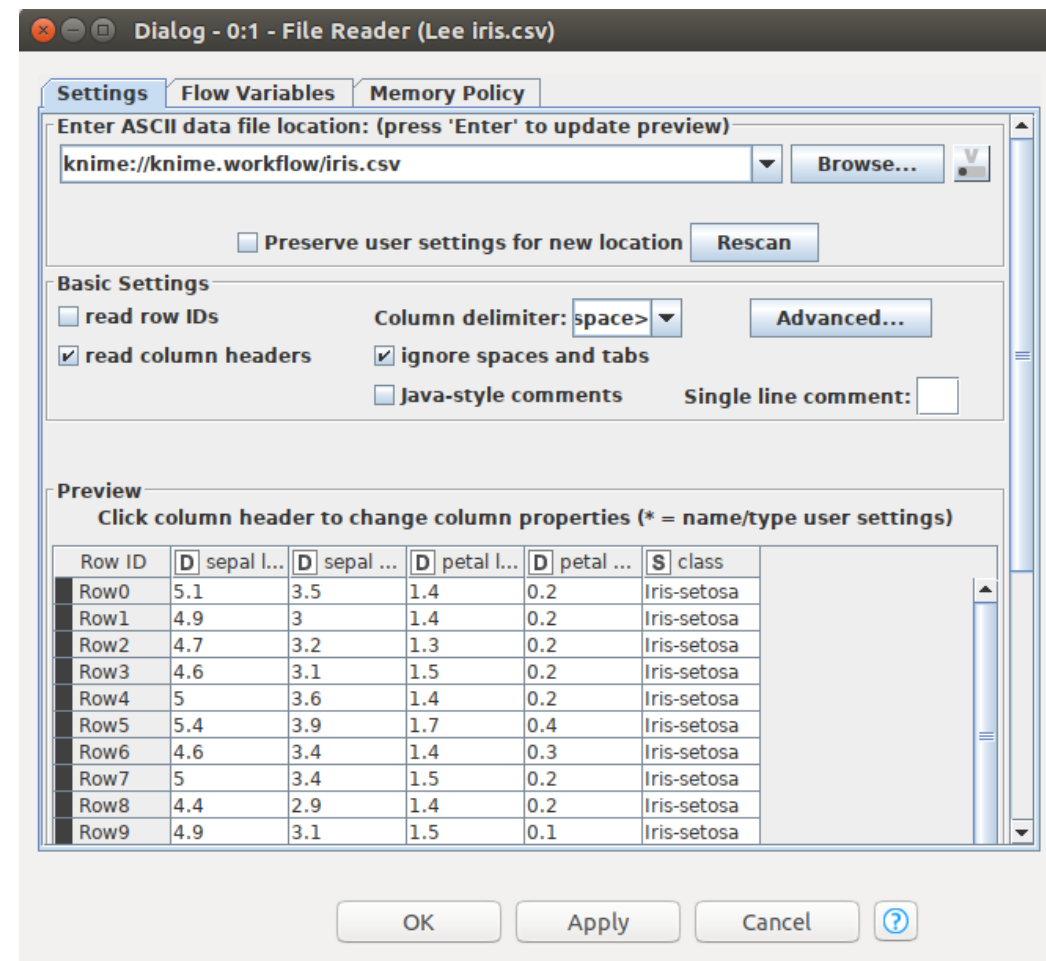


3. Ejemplo genérico: Clasificando especies de flores. Cargando datos

Podemos configurar el nodo haciendo doble click sobre el.

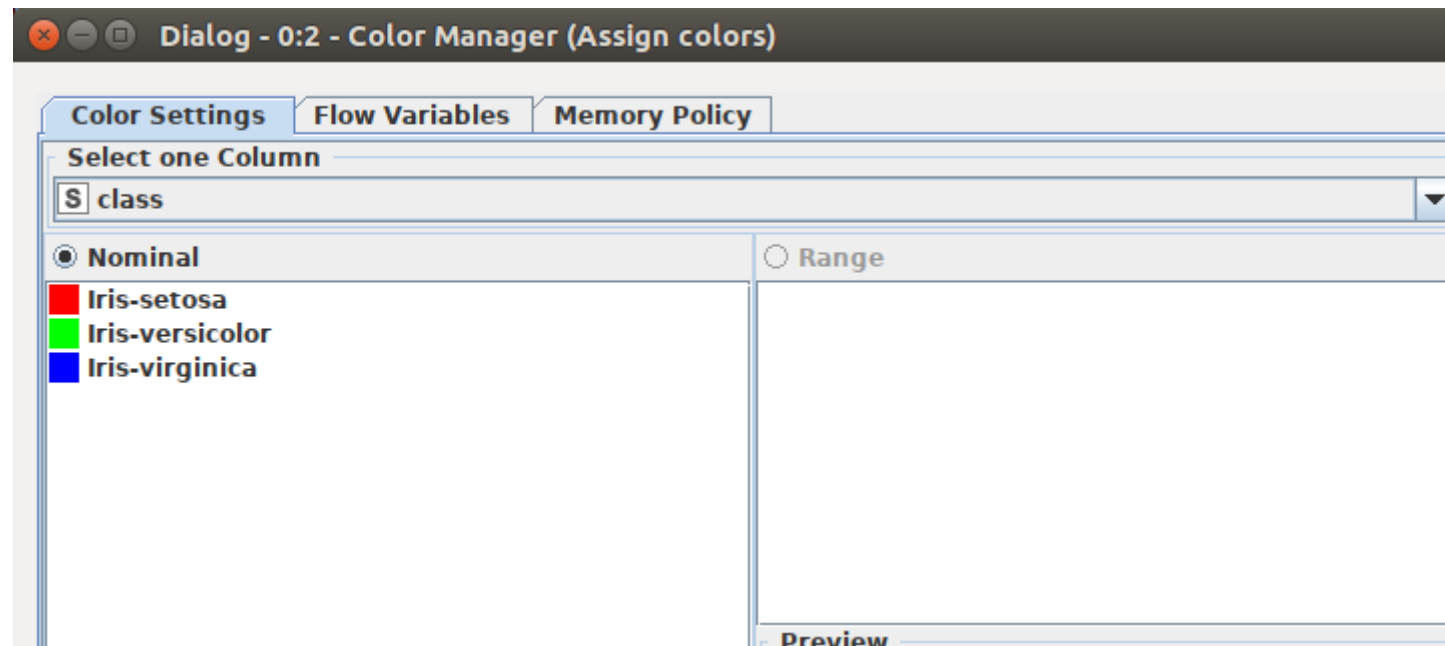
Al configurarlo, podemos establecer como es la cabecera o los delimitadores del fichero que queremos cargar.

Esto es necesario, porque a veces tenemos ficheros de datos separados por comas, otras veces punto y coma y otras veces tabuladores etc



3. Ejemplo genérico: Clasificando especies de flores. Coloreando los datos.

El nodo «Color Manager» nos permite colorear el conjunto de datos en función de los valores de uno de sus atributos.



3. Ejemplo genérico: Clasificando especies de flores. Coloreando los datos.

El nodo «Color Manager» nos permite colorear el conjunto de datos en función de los valores de uno de sus atributos.

El resultado es una tabla en la que cada fila está coloreada de acuerdo al valor del atributo que hemos elegido.

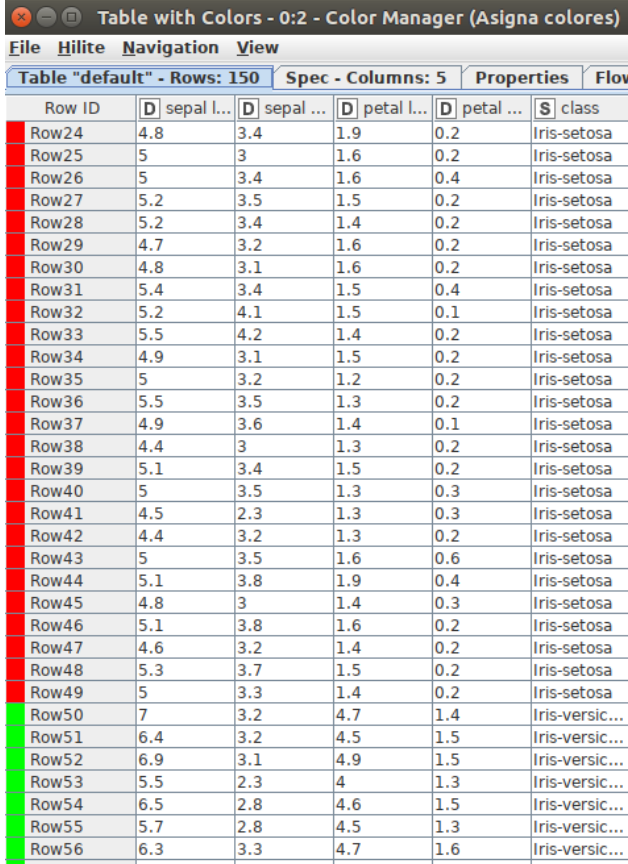


Table with Colors - 0:2 - Color Manager (Asigna colores)

File Hilite Navigation View

Table "default" - Rows: 150 Spec - Columns: 5 Properties Flow

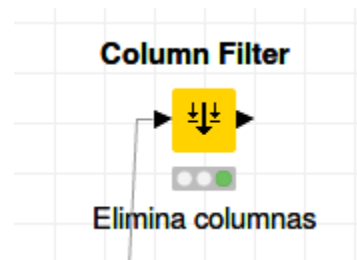
Row ID	D] sepal l...	D] sepal ...	D] petal l...	D] petal ...	S] class
Row24	4.8	3.4	1.9	0.2	Iris-setosa
Row25	5	3	1.6	0.2	Iris-setosa
Row26	5	3.4	1.6	0.4	Iris-setosa
Row27	5.2	3.5	1.5	0.2	Iris-setosa
Row28	5.2	3.4	1.4	0.2	Iris-setosa
Row29	4.7	3.2	1.6	0.2	Iris-setosa
Row30	4.8	3.1	1.6	0.2	Iris-setosa
Row31	5.4	3.4	1.5	0.4	Iris-setosa
Row32	5.2	4.1	1.5	0.1	Iris-setosa
Row33	5.5	4.2	1.4	0.2	Iris-setosa
Row34	4.9	3.1	1.5	0.2	Iris-setosa
Row35	5	3.2	1.2	0.2	Iris-setosa
Row36	5.5	3.5	1.3	0.2	Iris-setosa
Row37	4.9	3.6	1.4	0.1	Iris-setosa
Row38	4.4	3	1.3	0.2	Iris-setosa
Row39	5.1	3.4	1.5	0.2	Iris-setosa
Row40	5	3.5	1.3	0.3	Iris-setosa
Row41	4.5	2.3	1.3	0.3	Iris-setosa
Row42	4.4	3.2	1.3	0.2	Iris-setosa
Row43	5	3.5	1.6	0.6	Iris-setosa
Row44	5.1	3.8	1.9	0.4	Iris-setosa
Row45	4.8	3	1.4	0.3	Iris-setosa
Row46	5.1	3.8	1.6	0.2	Iris-setosa
Row47	4.6	3.2	1.4	0.2	Iris-setosa
Row48	5.3	3.7	1.5	0.2	Iris-setosa
Row49	5	3.3	1.4	0.2	Iris-setosa
Row50	7	3.2	4.7	1.4	Iris-versic...
Row51	6.4	3.2	4.5	1.5	Iris-versic...
Row52	6.9	3.1	4.9	1.5	Iris-versic...
Row53	5.5	2.3	4	1.3	Iris-versic...
Row54	6.5	2.8	4.6	1.5	Iris-versic...
Row55	5.7	2.8	4.5	1.3	Iris-versic...
Row56	6.3	3.3	4.7	1.6	Iris-versic...
Row57	4.9	3.4	3.3	1	Iris-versic...

3. Ejemplo genérico: Clasificando especies de flores. Eliminando columnas.

Column Filter: Es un nodo que permite elegir que columnas queremos excluir de los siguientes

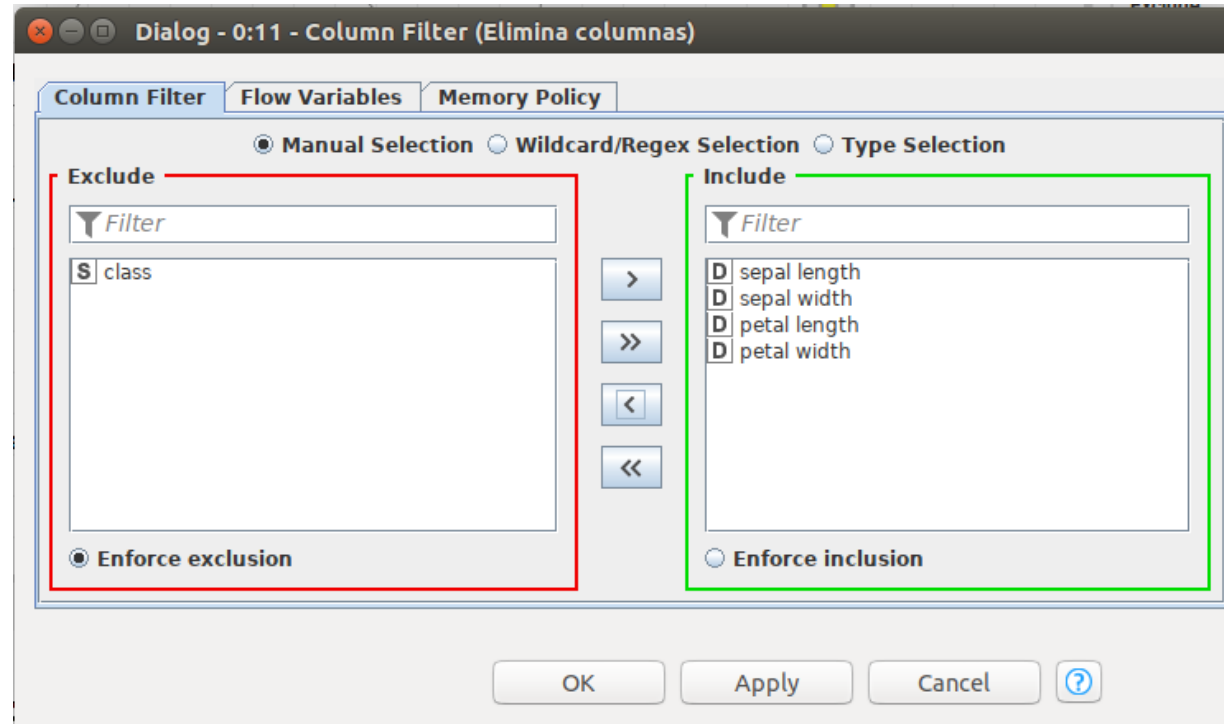
pasos del análisis.

En algunos caso puede ser necesario eliminar columnas porque tienen valores desconocidos o erróneos, en el ejemplo solamente vamos a borrar para ver que pasa.



3. Ejemplo genérico: Clasificando especies de flores. Eliminando columnas.

Hacemos doble click en el nodo y elegimos las columnas que se excluirán de los siguientes pasos.

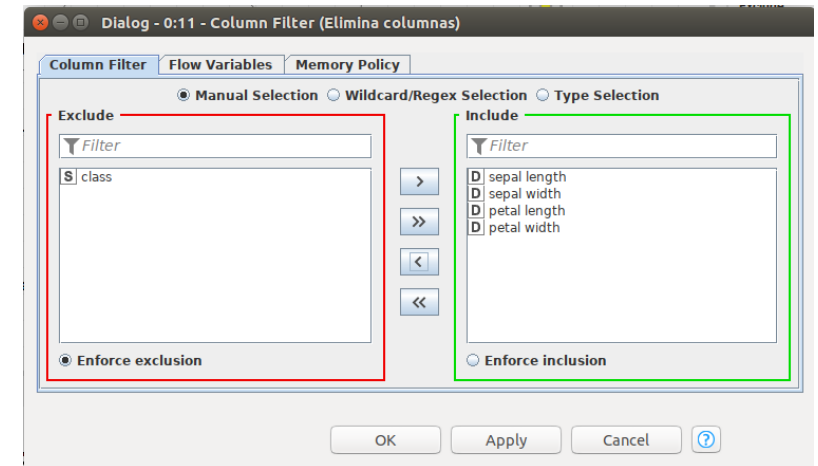


3. Ejemplo genérico: Clasificando especies de flores. Eliminando columnas.

En el ejemplo, se va a borrar la columna "Class", que es la que contiene el nombre de la especie a la que pertenece la flor descrita en cada ejemplo.

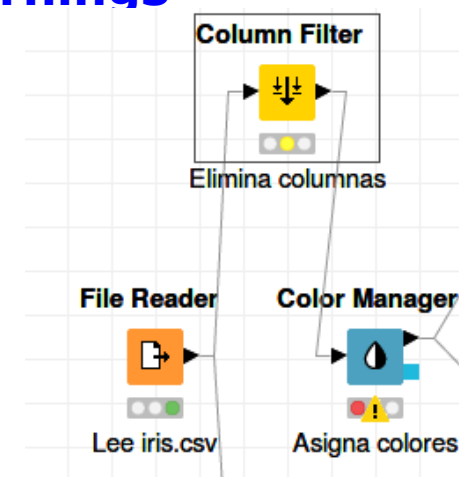
Solo vamos a hacer esto para provocar un error en el flujo de trabajo.

Saber identificar los tipos de error es fundamental para usar una herramienta como KNIME



3. Ejemplo genérico: Clasificando especies de flores. Errores y warnings

- Eliminamos la conexión entre «File Reader» y «Color Manager».
- Configuramos «Column Filter» para eliminar la clase.
- Conectamos «Column Filter» con «Color Manager»



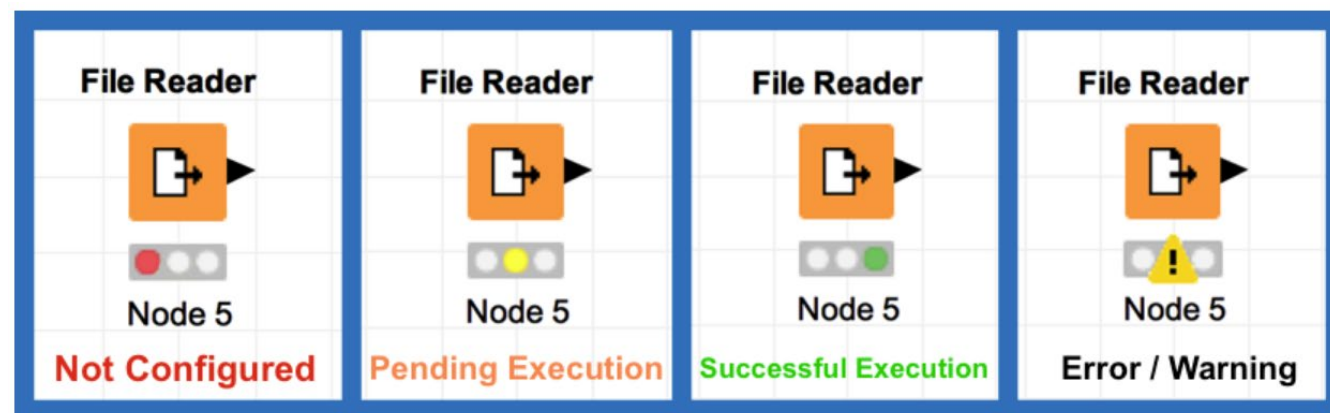
Ahora nos aparece un error en «Color Manager» porque este nodo utilizaba la clase para dar color a los ejemplos.

Para continuar, re-establecemos la conexión entre «File Reader» y «Color Manager»

3. Ejemplo genérico: Clasificando especies de flores. Warnings y errores.

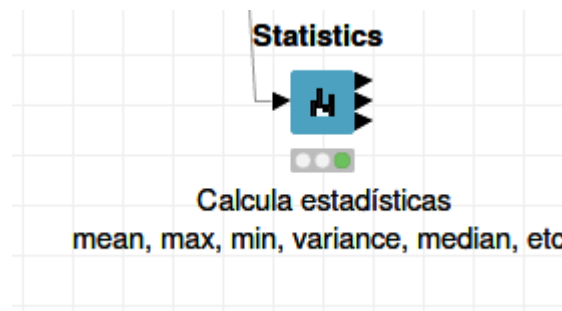
Un nodo puede estar en 4 estados diferentes:

- No configurado. Debemos hacer doble click sobre el y elegir algún parámetro importante que la herramienta no puede elegir por nosotros.
- Pendiente. Falta pulsar el botón de ejecución.
- Ejecutado.
- Error/ Warning. No se puede ejecutar. (como en el caso anterior, al borrar una columna usada por un nodo posterior)

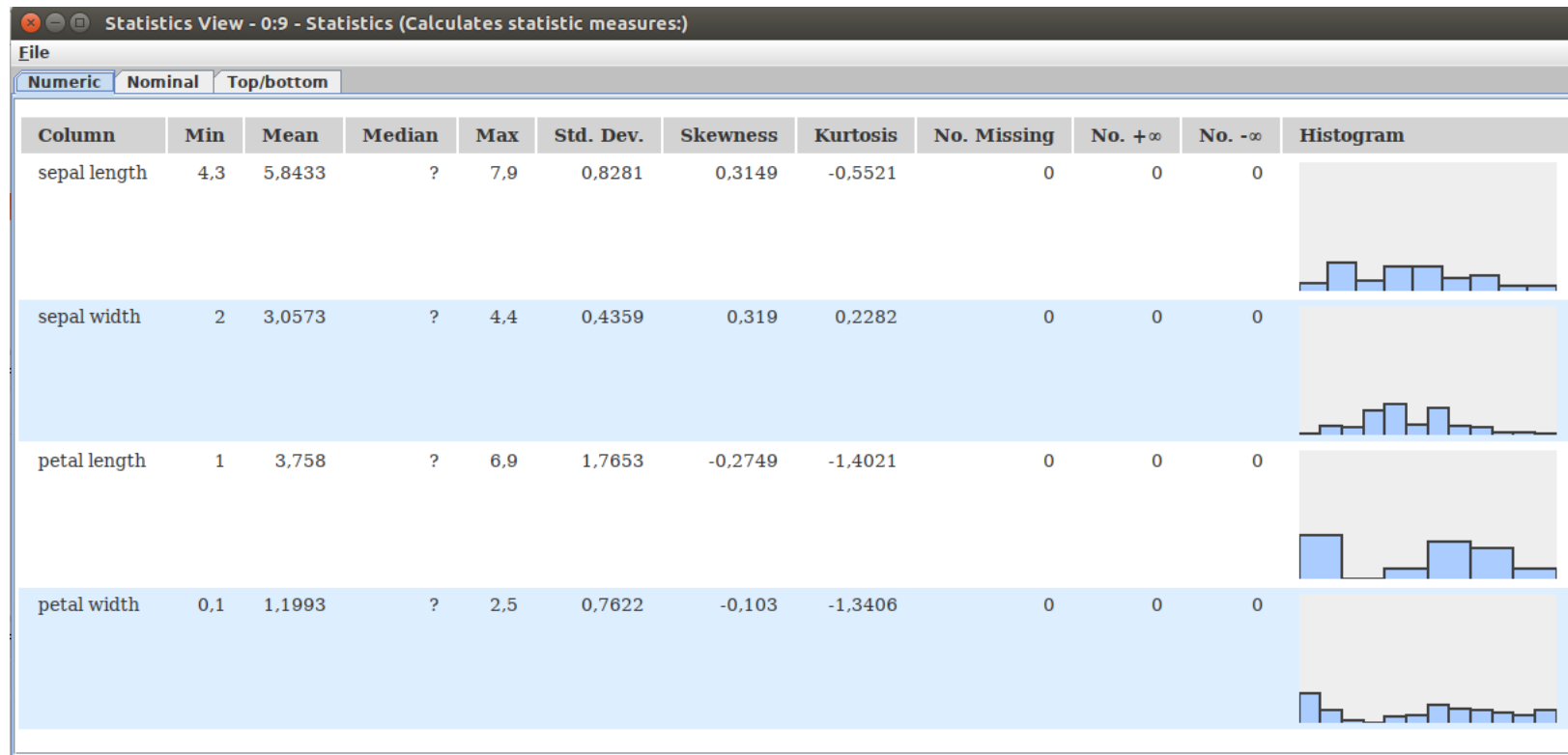


3. Ejemplo genérico: Clasificando especies de flores. Calcular estadísticas.

El nodo «Statistics», permite obtener estadísticas de una tabla de datos. Seleccionando el nodo y luego pulsando en «Statistics View» podemos sacar una tabla con estadísticas de cada uno de los atributos.

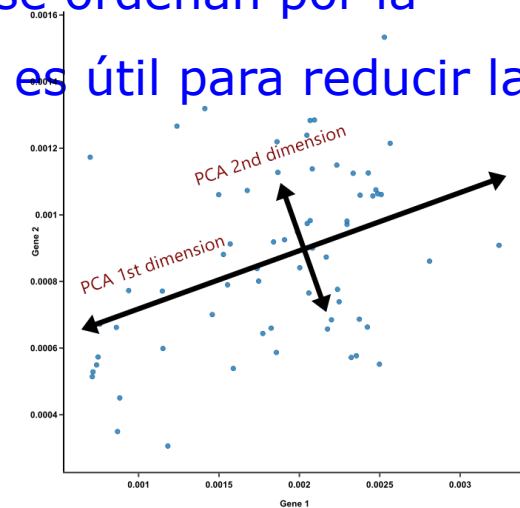
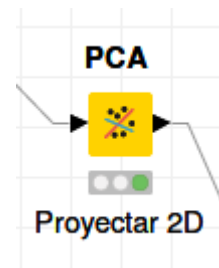


3. Ejemplo genérico: Clasificando especies de flores. Calcular estadísticas.



3. Ejemplo genérico: Clasificando especies de flores. Análisis de componentes principales.

PCA (Principal Component Analysis) o Análisis de componentes principales es una técnica estadística utilizada para describir un conjunto de datos en términos de nuevas variables, no correlacionadas, llamadas «componentes». Los componentes se ordenan por la cantidad de varianza original que describen, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos.

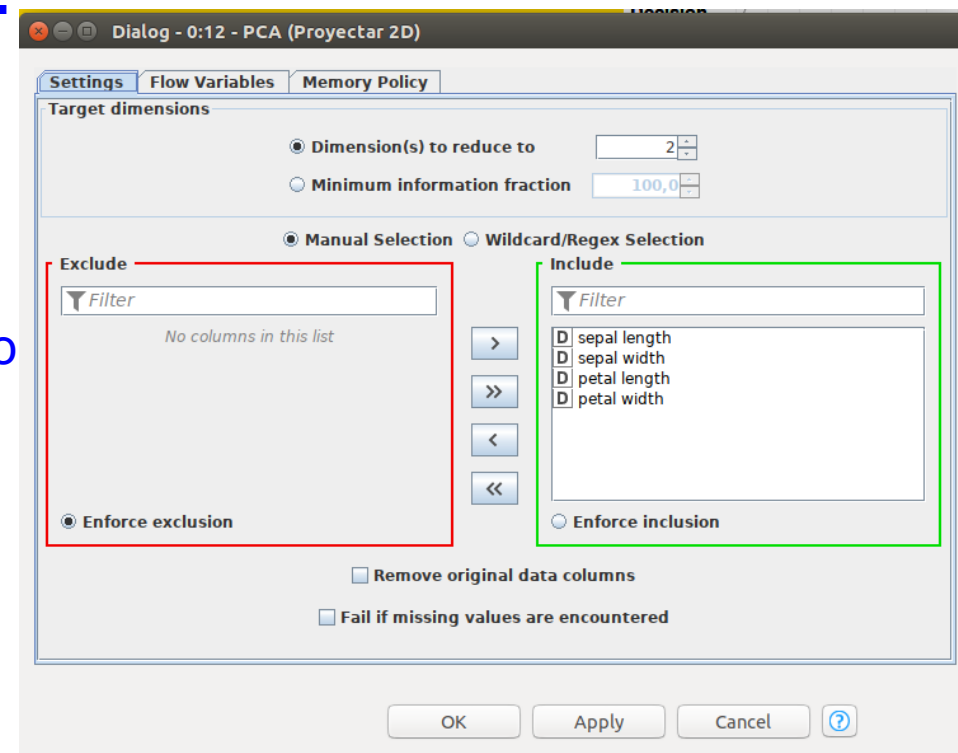


3. Ejemplo genérico: Clasificando especies de flores. Análisis de componentes principales.

Esta técnica nos ayuda a visualizar en 2D conjuntos de datos que tienen más de 2 atributos, y así podemos observar si hay «outliers», solapamiento entre las clases o si la frontera entre las clases es lineal o no lineal.

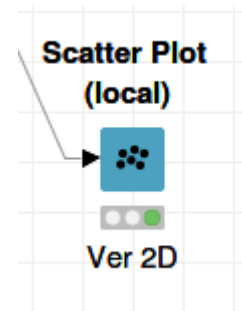
En el nodo se puede configurar cuantas componentes queremos calcular.

PCA crea nuevos atributos, no visualiza directamente, si queremos visualizar tenemos que conectar un nodo para hacer gráficos (lo veremos a continuación).



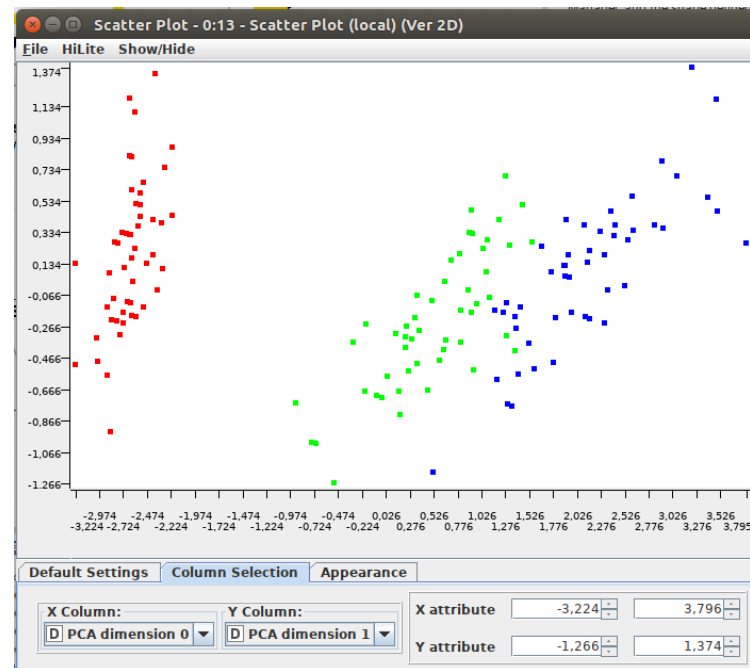
3. Ejemplo genérico: Clasificando especies de flores. Gráfico de dispersión.

El gráfico de dispersión nos permite visualizar dos atributos simultáneamente. Los ejemplos se representarán como puntos en un espacio 2D, en las coordenadas definidas por el valor de sus atributos.



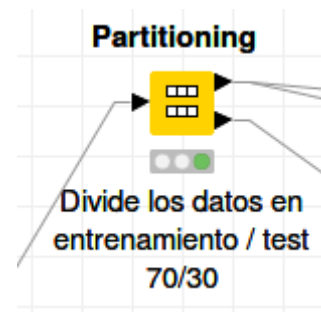
3. Ejemplo genérico: Clasificando especies de flores. Gráfico de dispersión.

En el ejemplo se están visualizando las dos primeras componentes principales, esta técnica permite resumir y visualizar (con dos atributos) un conjunto de datos que tenga varias columnas.



3. Ejemplo genérico: Clasificando especies de flores. Particionamiento de los datos.

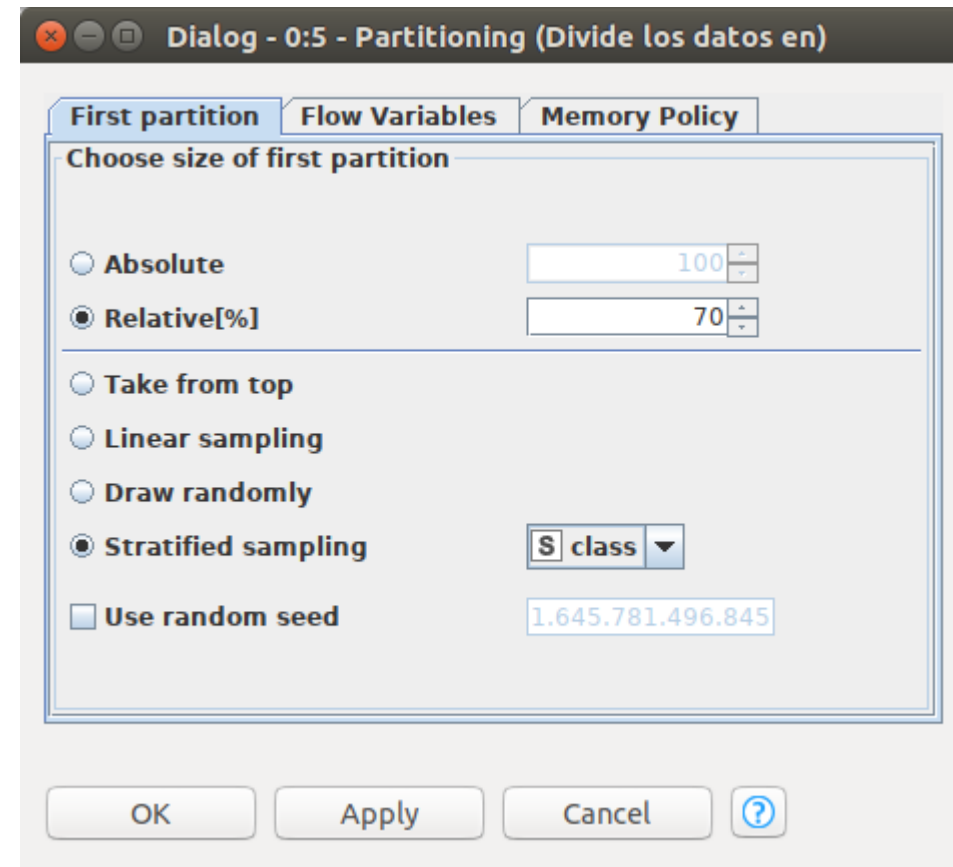
Para evaluar correctamente un algoritmo de minería de datos, debemos utilizar un conjunto distinto del usado para entrenar. A menudo disponemos de datos limitados, por lo que tenemos que dividir el conjunto de datos en parte de entrenamiento y parte de test.



3. Ejemplo genérico: Clasificando especies de flores. Particionamiento de los datos.

En el nodo podemos elegir el % de instancias que utilizaremos para entrenar y el % que usaremos para testear.

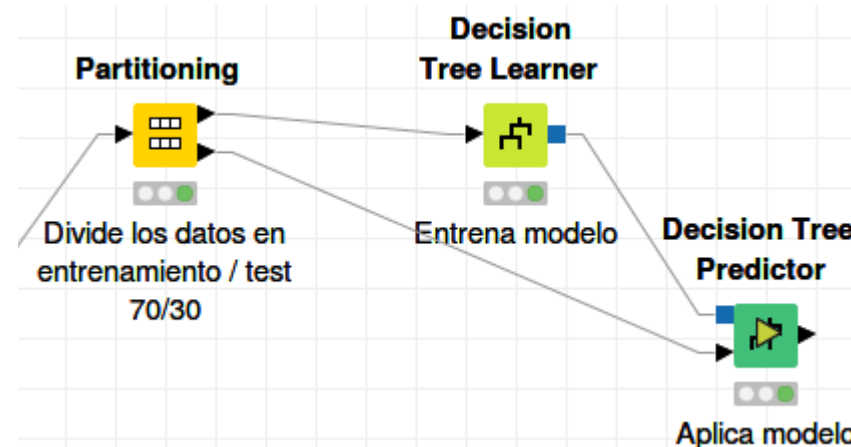
Hay opciones para hacer esta partición totalmente aleatoria o «estratificada» para que mantenga la proporción de clases tanto en entrenamiento como en test.



3. Ejemplo generico: Clasificando especies de flores. Creacion de modelos de minería de datos.

En KNIME tenemos múltiples algoritmos de aprendizaje. Suelen estar implementados mediante dos nodos:

- «Learner» Construye el modelo (entrena) a partir de los datos.
- «Predictor» Utiliza el modelo entrenado para predecir etiquetas de los datos nuevos que estén sin etiquetar o para predecir las etiquetas de los datos de test y evaluar su funcionamiento.



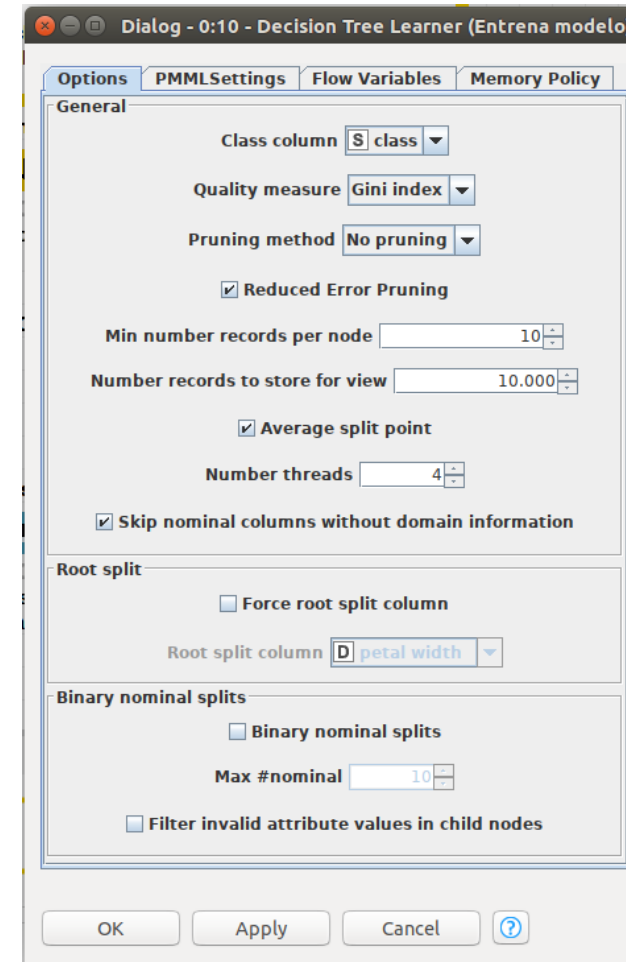
3. Ejemplo genérico: Clasificando especies de flores. Creación de modelos de minería de datos.

En KNIME tenemos múltiples algoritmos de aprendizaje. Suelen estar implementados mediante dos nodos:

- «Learner» Construye el modelo (entrena) a partir de los datos.
 - Da como resultado un modelo que en algunos casos se puede visualizar.
- «Predictor» Utiliza el modelo entrenado para predecir etiquetas de los datos nuevos.
 - Da como resultado una tabla, con una nueva columna correspondiente a las predicciones.

3. Ejemplo genérico: Clasificando especies de flores. Creación de modelos de minería de datos.

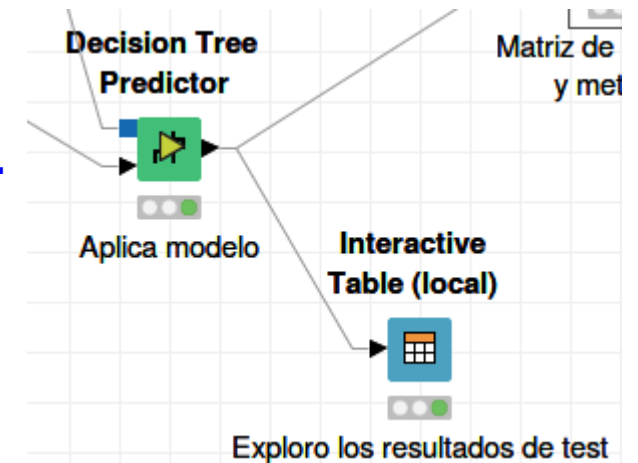
En el ejemplo se puede ver el interior del nodo «Learner» de KNIME de un árbol de clasificación. Permite configurar entre otras cosas, la medida de calidad de los atributos, si tiene poda o no poda etc.



3. Ejemplo genérico: Clasificando especies de flores. Visualizar resultados.

Podemos utilizar un nodo de tipo tabla interactiva para poder visualizar los valores de la clase real y la clase predicha, para todos los ejemplos de test.

De esta forma podemos ver los ejemplos clasificados erróneamente.



3. Ejemplo genérico: Clasificando especies de flores. Visualizar resultados.

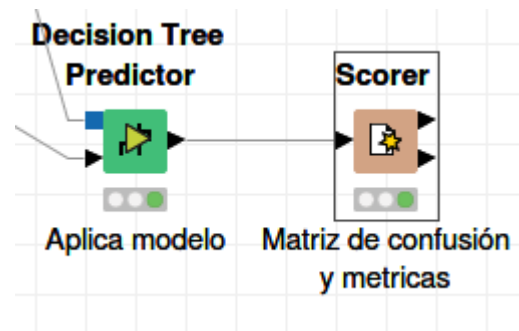
De esta forma podemos ver los ejemplos clasificados erróneamente.

Row ID	sepal l...	sepal ...	petal l...	petal ...	class	Prediction (class)
Row0	5.1	3.5	1.4	0.2	Iris-setosa	Iris-setosa
Row6	4.6	3.4	1.4	0.3	Iris-setosa	Iris-setosa
Row16	5.4	3.9	1.3	0.4	Iris-setosa	Iris-setosa
Row17	5.1	3.5	1.4	0.3	Iris-setosa	Iris-setosa
Row18	5.7	3.8	1.7	0.3	Iris-setosa	Iris-setosa
Row24	4.8	3.4	1.9	0.2	Iris-setosa	Iris-setosa
Row27	5.2	3.5	1.5	0.2	Iris-setosa	Iris-setosa
Row28	5.2	3.4	1.4	0.2	Iris-setosa	Iris-setosa
Row30	4.8	3.1	1.6	0.2	Iris-setosa	Iris-setosa
Row32	5.2	4.1	1.5	0.1	Iris-setosa	Iris-setosa
Row37	4.9	3.6	1.4	0.1	Iris-setosa	Iris-setosa
Row39	5.1	3.4	1.5	0.2	Iris-setosa	Iris-setosa
Row40	5	3.5	1.3	0.3	Iris-setosa	Iris-setosa
Row44	5.1	3.8	1.9	0.4	Iris-setosa	Iris-setosa
Row46	5.1	3.8	1.6	0.2	Iris-setosa	Iris-setosa
Row52	6.9	3.1	4.9	1.5	Iris-versicolor	Iris-virginica
Row59	5.2	2.7	3.9	1.4	Iris-versicolor	Iris-versicolor
Row60	5	2	3.5	1	Iris-versicolor	Iris-versicolor

3. Ejemplo genérico: Clasificando especies de flores. Visualizar resultados.

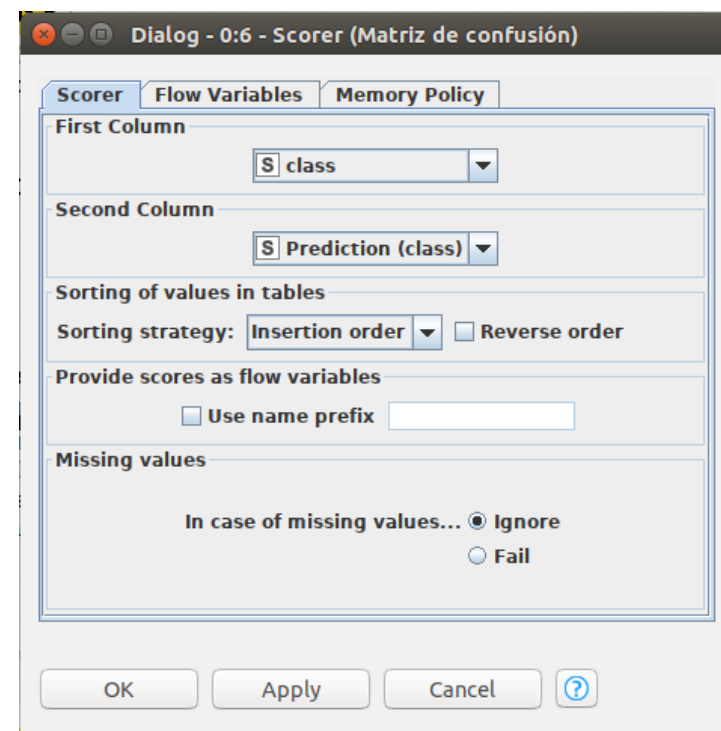
Además de visualizar las predicciones podemos evaluar fácilmente la calidad de estas predicciones.

El nodo «Scorer» se utiliza para evaluar los resultados. Se puede utilizar para obtener medidas como la tasa de acierto y obtener la matriz de confusión.



3. Ejemplo genérico: Clasificando especies de flores. Visualizar resultados.

Para configurar el nodo «Scorer» debemos definir cual es la columna que representa la clase real y cual es la columna para la clase predicha por el modelo.



3. Ejemplo genérico: Clasificando especies de flores. Visualizar resultados.

En el ejemplo concreto (conjunto de datos del iris, usando un árbol de clasificación, 70% de los datos para entrenar y el 30% restante para evaluar) muestra la tasa de acierto global del modelo sobre el conjunto de datos de test (91 %) y otras métricas como falsos positivos o falsos negativos para cada una de las clases.

Accuracy statistics - 0:6 - Scorer (Matriz de confusión)

File Hilite Navigation View

Table "default" - Rows: 4 Spec - Columns: 11 Properties Flow Variables

Row ID	TruePo...	FalseP...	TrueN...	FalseN...	Recall	Precision	Sensiti...	Specifity	F-mea...	Accur...	Cohen'...
Iris-setosa	15	0	30	0	1	1	1	1	1	?	?
Iris-versicolor	12	1	29	3	0.8	0.923	0.8	0.967	0.857	?	?
Iris-virginica	14	3	27	1	0.933	0.824	0.933	0.9	0.875	?	?
Overall	?	?	?	?	?	?	?	?	?	0.911	0.867

3. Ejemplo genérico: Clasificando especies de flores.

Visualizar resultados.

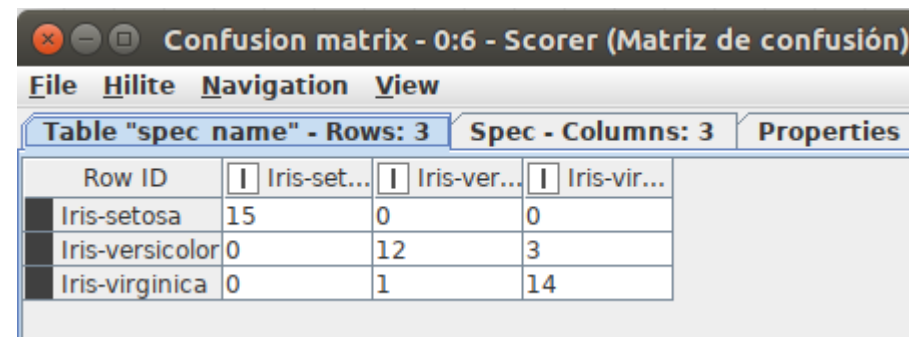
En el mismo nodo también podemos obtener la matriz de confusión, que relaciona la clase real con la clase predicha, para observar los tipos de errores del modelo.

En dicha tabla:

- Filas: Son las Clases reales.
- Columnas: Son las Clases predichas.

En el ejemplo:

- 3 ejemplos de iris-versicolor fueron clasificados erróneamente como iris-virgínica.
- 1 ejemplo de iris-virgínica fue clasificado erróneamente como iris-versicolor.

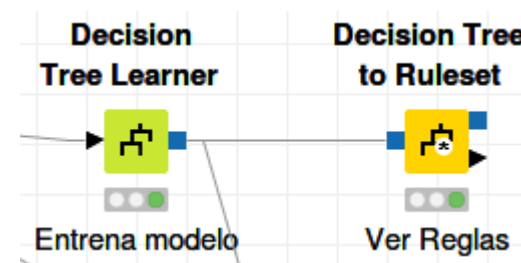
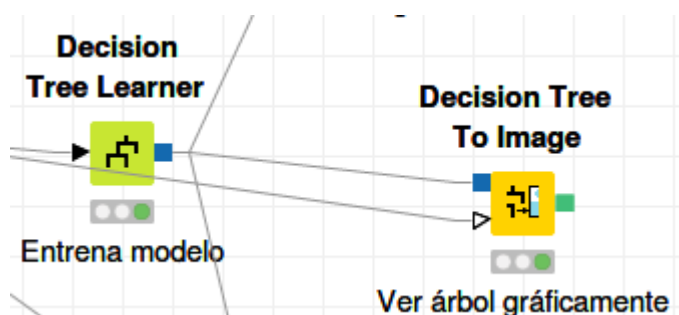


Row ID	Iris-set...	Iris-ver...	Iris-vir...
Iris-setosa	15	0	0
Iris-versicolor	0	12	3
Iris-virginica	0	1	14

3. Ejemplo genérico: Clasificando especies de flores. Visualizar el modelo.

Algunos modelos de minería de datos son interpretables, es decir, podemos interpretar el cómo llega a una conclusión, como clasifica un determinado ejemplo como una clase y no como la clase contraria.

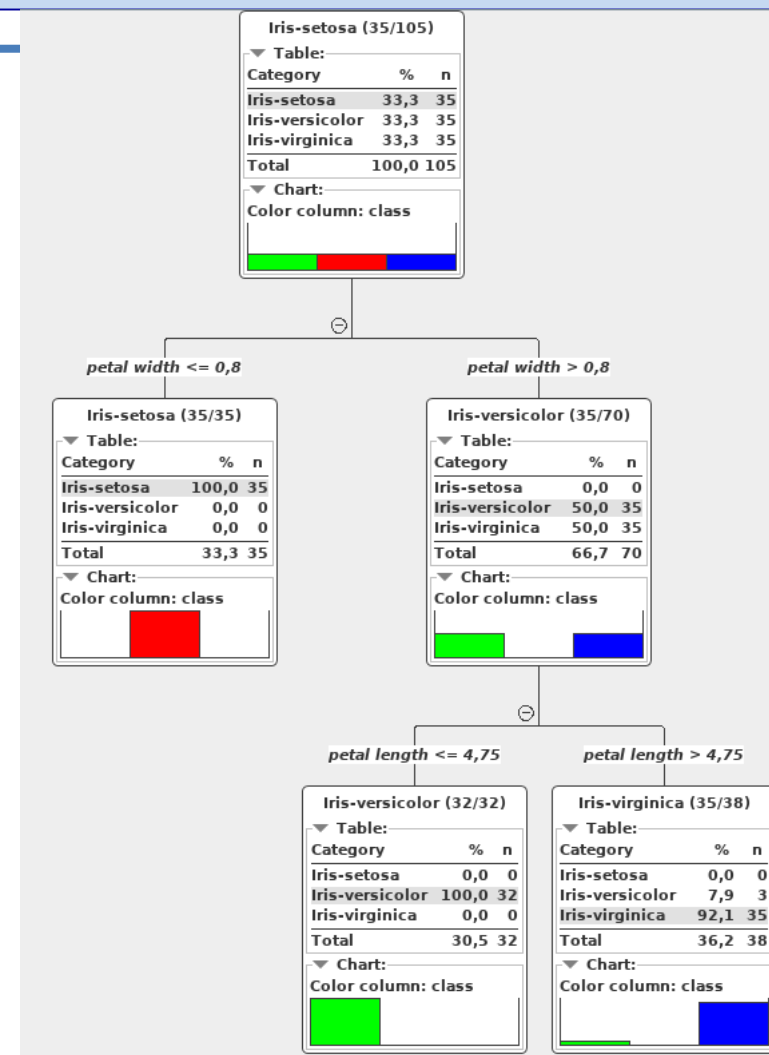
En el caso concreto de los árboles podemos verlos en forma gráfica o en forma de reglas, si son demasiado grandes como para interpretarlos correctamente en su forma gráfica.



3. Ejemplo genérico: Clasificando especies de flores. Visualizar el modelo.

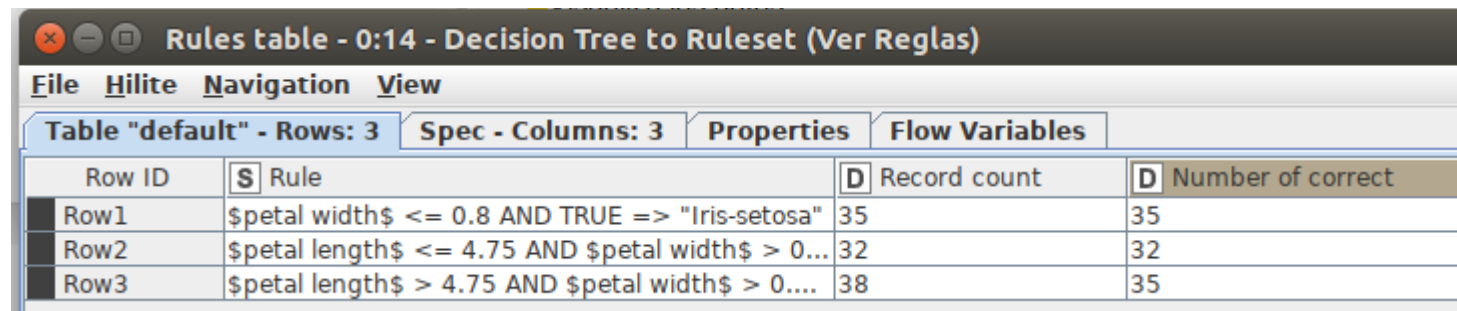
Visualizando el árbol de clasificación.

- Si la anchura del pétalo es menor de 0.8 clasificamos el ejemplo como setosa.
- Si no
 - Si la longitud del pétalo es menor de 4.75 clasificamos el ejemplo como versicolor.
 - Si no, clasificamos el ejemplo como virgínica.



3. Ejemplo genérico: Clasificando especies de flores. Visualizar el modelo.

Un árbol más grande no sería práctico visualizarlo en forma gráfica, así que puede traducirse a un conjunto de reglas que es un representación más compacta.



Row ID	Rule	Record count	Number of correct
Row1	\$petal width\$ <= 0.8 AND TRUE => "Iris-setosa"	35	35
Row2	\$petal length\$ <= 4.75 AND \$petal width\$ > 0...	32	32
Row3	\$petal length\$ > 4.75 AND \$petal width\$ > 0....	38	35

4. Ejemplo con datos de intervención terapéutica inteligente (EaryCare).

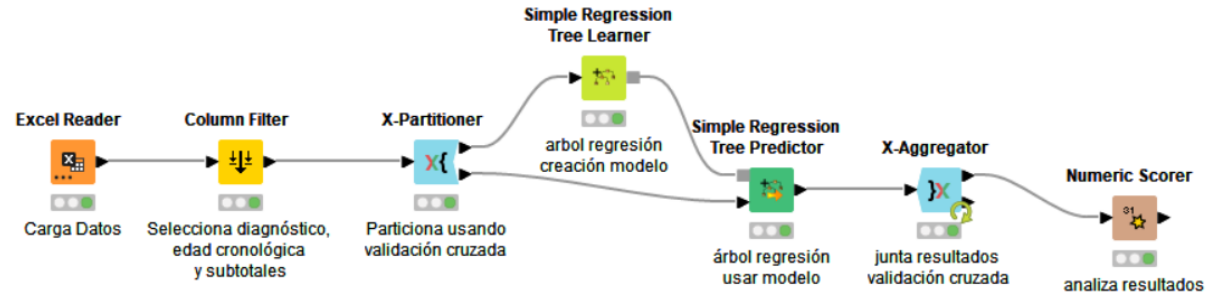
- Se necesita descargar el fichero llamado «eEarlyCare.knwf»
- En el Explorador de KNIME hacer click derecho y después «Import KNIME workflow ...»
- Posteriormente la opción «Select file» → «browse»
- Lo seleccionamos y damos «Ok»

Material Adicional: Usando KNIME

4. Ejemplo con datos de intervención terapéutica inteligente (EaryCare).

Es un flujo de trabajo que utiliza un conjunto de datos que se utilizan como variables independientes los ítems de la escala eEarlyCare, la edad cronológica y el sexo, y como variable dependiente el diagnóstico principal.

Explora el workflow.



Ejemplo de regresión con los datos de eEarlyCare

Web

KNIME → <https://www.knime.com/>



¡¡¡MUCHAS GRACIAS POR
VUESTRA ATENCIÓN!!!



Co-funded by
the European Union

