



INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

SITUACIÓN PROFESIONAL 2

CLASE 3

Prof. Ricardo Piña

Ver en Video:

Título: [IA] Clase 3

link: <https://www.youtube.com/watch?v=YvbFfM3OIU8&t=1237s> (<https://www.youtube.com/watch?v=YvbFfM3OIU8&t=1237s>)

desde: inicio

hasta: 34:43

Out[2]:

A esta altura de la carrera Ud. todavía no sabe programar en Python, así que en este archivo hemos ocultado las celdas que contienen código para facilitar su lectura. Si Ud. quiere ver el código u ocultarlo, haga [click aquí](#).

Vamos a resolver el problema planteado al inicio de la Situación Profesional 2 referido al problema del otorgamiento del crédito bancario, pero ahora lo resolveremos de una forma más cómoda: con software.

Out[3]:

SOFTWARE QUE USAREMOS EN ESTA MATERIA

Como mientras cursas la materia Introducción a la Inteligencia Artificial estarás dando los primeros pasos en programación con la materia Programación I no estaremos en condiciones de elaborar programas complejos en ningún lenguaje, por lo cual utilizaremos un programa que corre bajo Python y que nos evitará tener que programar: **Orange 3**.

Orange3 presenta una interfaz gráfica (GUI: Graphic User Interface) sencilla de usar y si bien no puede tener la versatilidad de lenguajes de programación como R o Python "para hacer lo que uno quiera" es muy adecuado para comenzar a aplicar métodos de Machine Learning.

INSTALACIÓN DE ORANGE 3

Orange3 se puede usar tanto para Windows como Linux y Mac. Ahora explicaremos la forma que te **recomendamos** para instalarlo, no es la única pero tiene sus ventajas. Los pasos serán los siguientes:

- Primero: instalar **Anaconda**.
- Segundo: instalar Orange 3 desde dentro de Anaconda.

Instalación de Anaconda

Anaconda es una **plataforma muy completa para las tareas de Data Science y Machine Learning** basada en Python, es decir que reúne muchos de los principales programas que vamos a usar durante toda la carrera, como habrás adivinado entre ellos cuales se encuentra Orange3. La versión para empresas es paga, pero para individuos es gratuita.

Para instalar Anaconda debes ir al sitio:

<https://www.anaconda.com/> (<https://www.anaconda.com/>)

y buscar la versión gratuita que es: **Anaconda Distribution**.

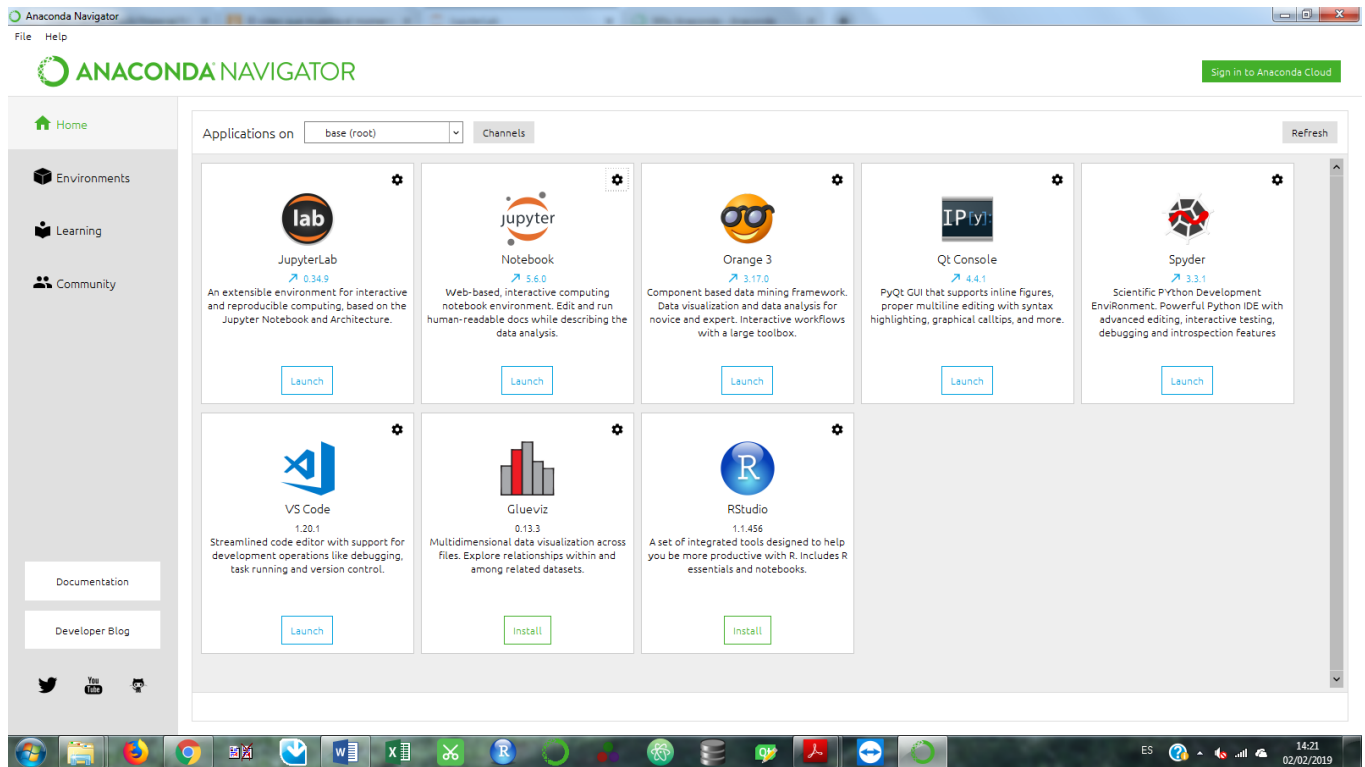
Mientras baja el instalador a tu máquina te recomendamos que "des una vuelta" por el sitio de Anaconda, más que nada por los vínculos relaciondos a **AI (Artificial Intelligence) y Data Science** ya que esa será tu profesión cuando egreses!

La instalación de Anaconda incluye la de Python, es decir que no deberás instalarlo aparte. Con respecto a Python hubo un cambio muy grande hace unos años de la versión 2.7 a las versiones 3.x que no son del todo compatibles. Si en algún momento te pregunta qué versión de Python deseas instalar elige la 3.x.

Una vez que Anaconda se instale correctamente verás que a través del botón inicio de Windows en la lista de programas debe figurar Anaconda con varios ítems, los más interesantes son:

- Anaconda Prompt. Permite usar Anaconda desde la línea de comandos (como si fuera una venta del viejo DOS o el PowerShell de Windows). Si bien no lo usaremos por ahora, en otras materias deberás usarlo, así que recuerda dónde está.
- Anaconda Navigator: presenta una interfaz gráfica para Anaconda y permite hacer las mayorías de las cosas que se hacen desde Anaconda prompt pero en forma visual.

Abre **Anaconda Navigator** y se presentará su interfaz gráfica:



Instalación de Orange 3

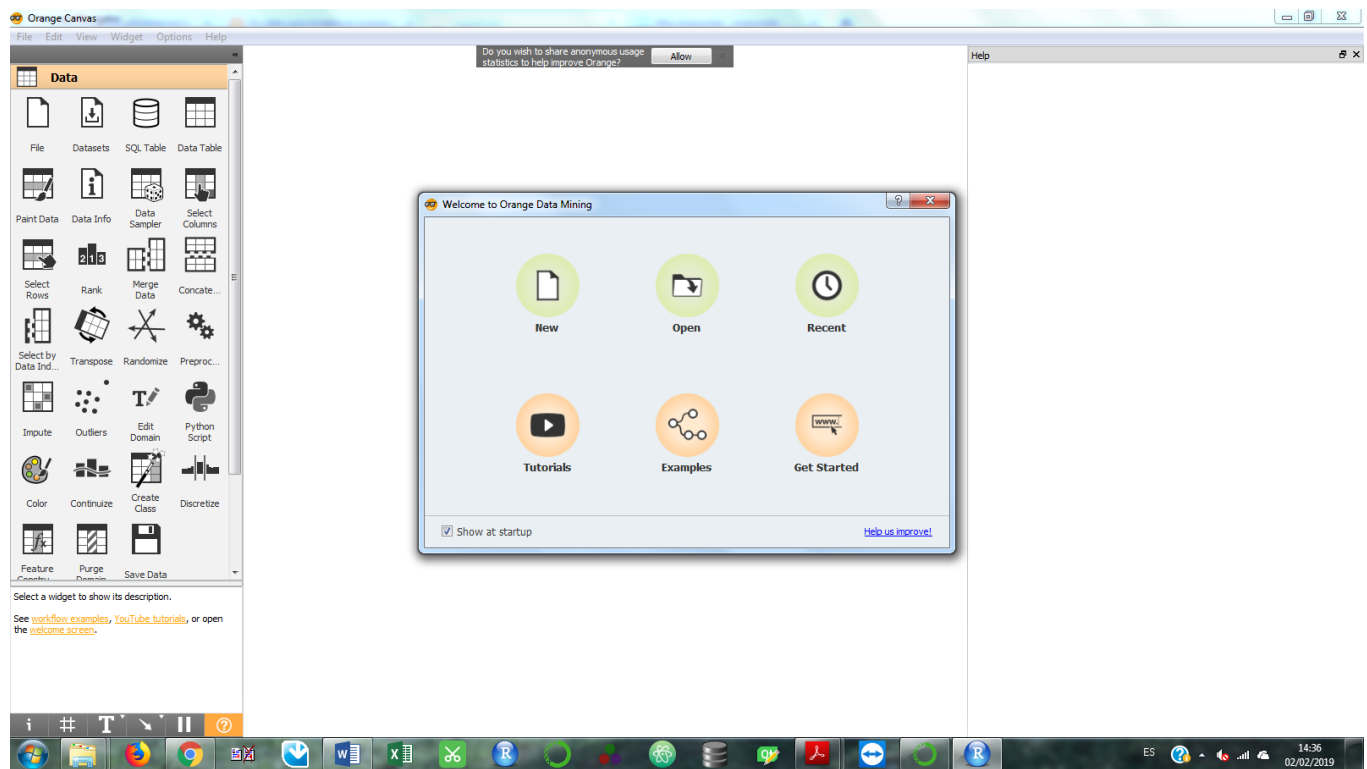
Como habíamos comentado Anaconda trae varios programas útiles para Data Science. En cada uno de los iconos debe figurar la opción:

- **Launch:** si ya está instalado y queremos "lanzarlo" o "abrirlo" para usarlo
- **Install:** si aún debe ser instalado

La instalación de cualquiera de estos programas es automática, Anaconda se comunicará por internet con los sitios donde residen y procederá a instalarlos (el programa instalador de Anaconda se llama **conda** y siempre que sea posible es preferente instalar los programas desde Anaconda / conda en vez de hacerlo directamente desde Python usando el instalador de Python que se llama **pip**)

- Si no está instalado Orange 3 haz clic en Install.
- Te sugerimos instalar también JupyterLab, y Spyder ya que los usarás en otras materias en este semestre, sino puedes instalarlos luego.

Una vez instalado Orange 3 puedes "lanzarlo" o "abrirlo" y verás una interfaz similar a la siguiente:



Haz click en **New** y accederás a la plataforma de trabajo para encargar nuestro proyecto.

Resolvamos el Problema de la Situación Profesional 2

Recordemos los datos del problema:

Out[4]:

	Tiene_Deuda	Genero	Trabaja	Propietario	Dar_Credito
0	Si	F	No	Si	No
1	Si	M	No	Si	No
2	No	F	Si	Si	Si
3	No	F	Si	No	Si
4	No	F	No	No	No
5	No	M	Si	No	No
6	No	M	No	Si	No
7	No	M	No	No	No
8	Si	F	No	No	No
9	Si	M	Si	No	No
10	Si	M	No	No	No
11	No	F	No	Si	No
12	No	M	Si	Si	Si

Antes que nada observemos que éste es un problema de Clasificación ya que pretendemos pronosticar los valores de Dar_Credito que pueden ser Si o No, los Árboles de Decisión son muy útiles para Clasificar.

Mientras usa Orange3 le recomendamos que utilice la documentación, en este link

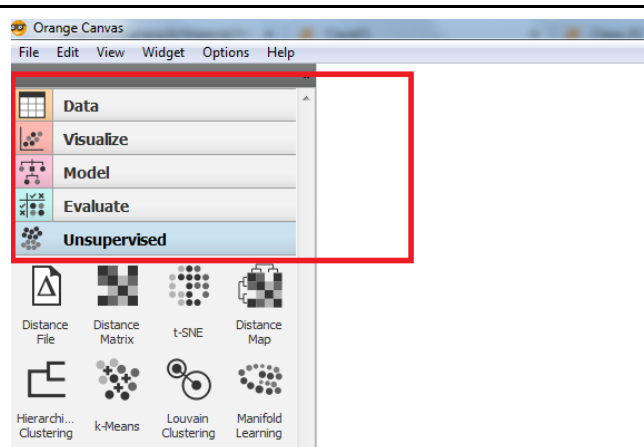
<https://media.readthedocs.org/pdf/orange-visual-programming/latest/orange-visual-programming.pdf> (<https://media.readthedocs.org/pdf/orange-visual-programming/latest/orange-visual-programming.pdf>) está la correspondiente a la interfaz visual de Orange, este archivo en formato pdf lo encontrará también en el carpeta docs.

Vamos a cargar estos datos en Orange

Carga de los Datos

Imagen del Panel de la Izquierda

Instrucciones



Observe que el Panel de la Izquierda presenta varias opciones: Data, Visualize, Model, Evaluate, Unsupervised

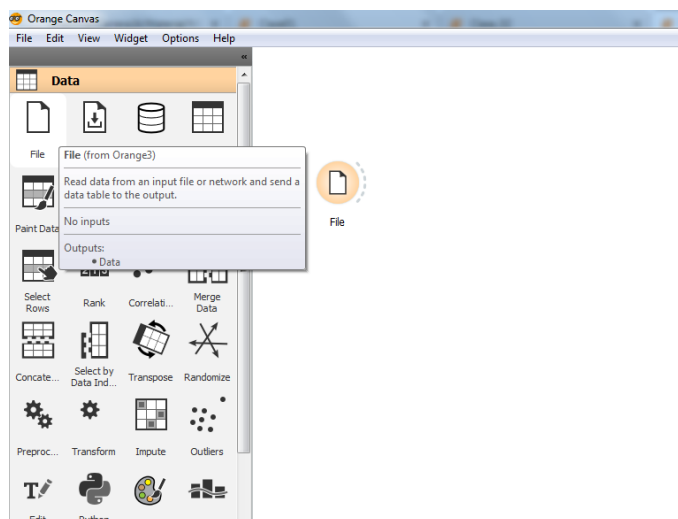
- Data: herramientas para abrir y guardar archivos, mostrar tablas de datos, procesar datos, etc
- Visualize: herramientas para visualizar relaciones entre datos, salidas de datos, etc
- Model: aquí están los distintos Modelos de Machine Learning, entre otros los Árboles de Decisión.
- Evaluate: herramientas para evaluar qué tan bien funciona el modelo que estamos probando
- Unsupervised: herramientas para aprendizaje no supervisado.

File

Como deseamos abrir un archivo con datos, encontraremos el objeto **File** en el panel Data:

Usar el Widget File

Instrucciones

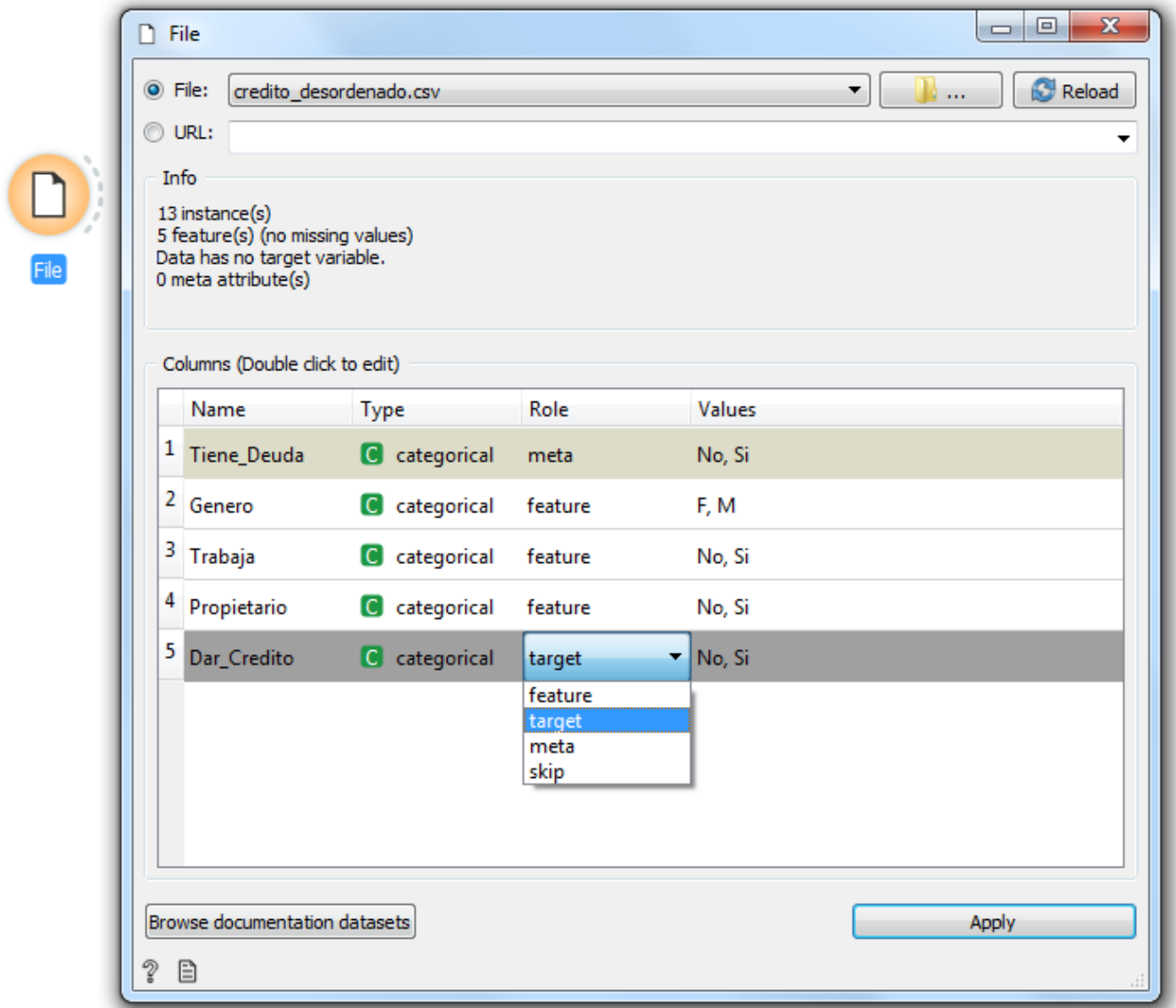


Observe que en el Panel de la izquierda hay un **widget** denominado **File**, arrástrelo hacia la derecha a la zona de trabajo de Orange.

Haga doble click sobre el objeto File y se abrirá una ventana, en la parte superior podrá buscar el archivo donde tenemos los datos. Navegue hasta que encuentre el archivo que necesitamos que es `credito_clase.csv` o bien `credito_desordenado.csv`

Orange3 permite abrir diversos tipos de archivos como csv, tab, archivos de Excel: xls, xlsx, archivos de Open Office, etc.

El control abrirá el archivo y le presentará un resumen denominado **"Info"**.



Observe también que le presenta las **características** de las columnas del archivo de datos.

De interés son las columnas:

- **Type:** al hacer doble clic sobre el "tipo" asignado a una variable nos mostrará los tipos posibles, en este caso nos presenta sólo **categorical** o **text**. Algo que aprenderá a lo largo de la carrera es que no hay un criterio uniforme para los tipos de variables, y en cada herramienta que utilicemos deberemos indagar cómo han definido a cada uno. Generalmente el tipo **texto** se utiliza por ejemplo para nombres, direcciones, respuestas abiertas a encuestas; en cambio el tipo de variable **categorical** significa que la variable asume unos pocos valores como Si/No; Alto/Bajo, Grande/Mediano/Pequeño. En otros contextos se distinguen las variables categóricas en **ordinales** lo cual significa que hay un orden entre los valores de las variables, por ejemplo: Primero/Segundo/Tercero en el caso de una competencia o Grande/Mediano/Pequeño y **no ordinales** como por ejemplo abierto/cerrado, visto/no visto, Si/No, etc Siempre hay que ver el contexto en el que están definidas. Como podemos observar Orange no toma en cuenta si son ordinales o no, así que las dejamos como **categorical**.
- **Role:** se refiere justamente al rol que tiene cada una de las variables en el conjunto de datos. Nos permite asignar a cada una de las variables en una de cuatro opciones:
 - **skip (saltar):** la columna no será usada
 - **target (objetivo):** la o las variables que pretendemos **pronosticar**
 - **feature (característica):** la columna corresponde a una variable que pensamos que contiene información útil con la que pretendemos pronosticar la variable target.
 - **meta (metadatos):** corresponde a datos adicionales o de contexto que no se usarán para procesar o analizar.

En nuestro caso a la variable Dar_Credito le tenemos que asignar el rol de target y las restante tendrán el rol de features.

Table

Ver los datos en forma de Tabla

Para ver el detalle de los datos en forma de tabla, debemos:

- arrastrar el widget **Data Table** hacia la zona de trabajo cerca de File
- luego conectar la salida de File con la Entrada del Data Table, arrastrando desde el arco punteado de la salida de File hasta la entrada punteada de la entrada de Data Table.
- finalmente haremos doble clic en Data Table para abrirlo y veremos nuestros datos.

The screenshot shows the 'Data Table' application window. On the left is a sidebar with settings, and on the right is a data table.

File

Data

Data Table

Info

- 13 instances (no missing values)
- 4 features (no missing values)
- Discrete class with 2 values (no missing values)
- No meta attributes

Variables

- ☒ Show variable labels (if present)
- ☐ Visualize numeric values
- ☒ Color by instance classes

Selection

- ☒ Select full rows

Restore Original Order

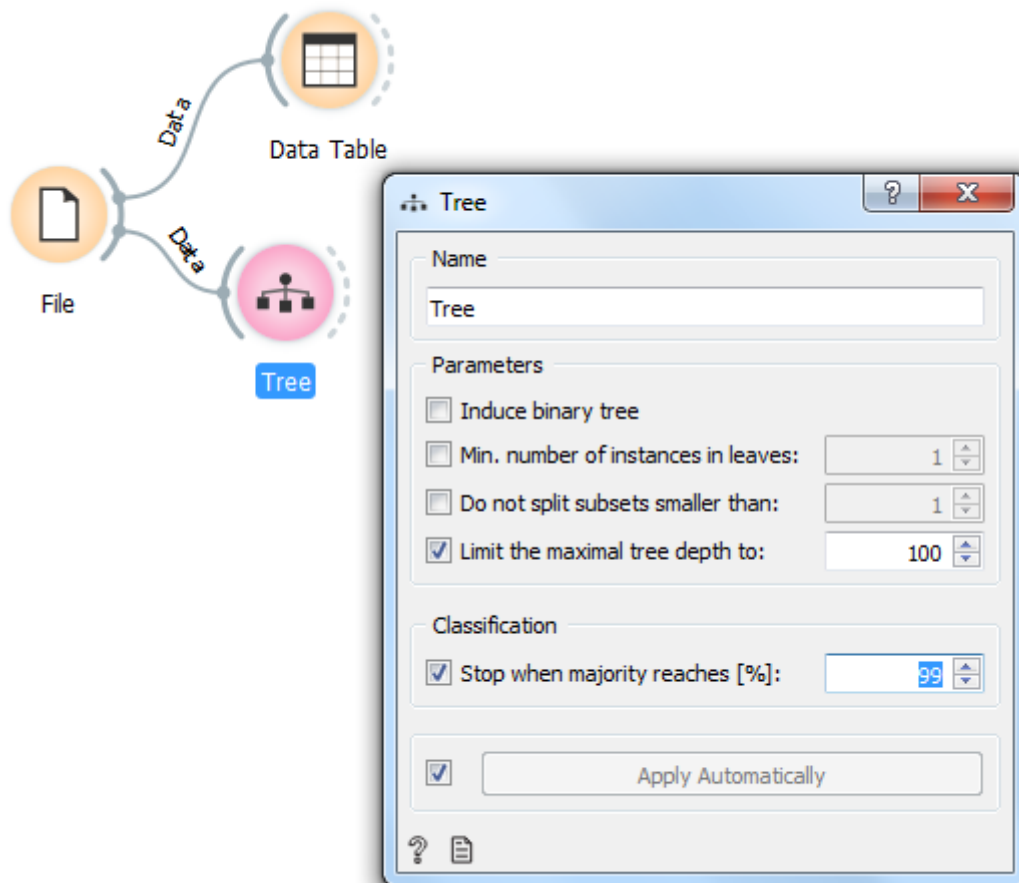
☒ **Send Automatically**

	Dar_Credito	Tiene_Deuda	Genero	Trabaja	Propietario
1	No	Si	F	No	Si
2	No	Si	M	No	Si
3	Si	No	F	Si	Si
4	Si	No	F	Si	No
5	No	No	F	No	No
6	No	No	M	Si	No
7	No	No	M	No	Si
8	No	No	M	No	No
9	No	Si	F	No	No
10	No	Si	M	Si	No
11	No	Si	M	No	No
12	No	No	F	No	Si
13	Si	No	M	Si	Si

Tree: Creamos el Modelo de Árbol de Decisión

Los Árboles de Decisiones son uno de los Modelos más tradicionales de Machine Learning, para aplicarlo:

- En el panel de la izquierda seleccione **Models**
- Arrastre el widget **Tree** hacia el área de trabajo
- Conecte desde la **salida de File -> entrada de Tree** (Observe que no se cargan los datos desde el Data Table sino desde File)
- Haga doble clic en Tree:

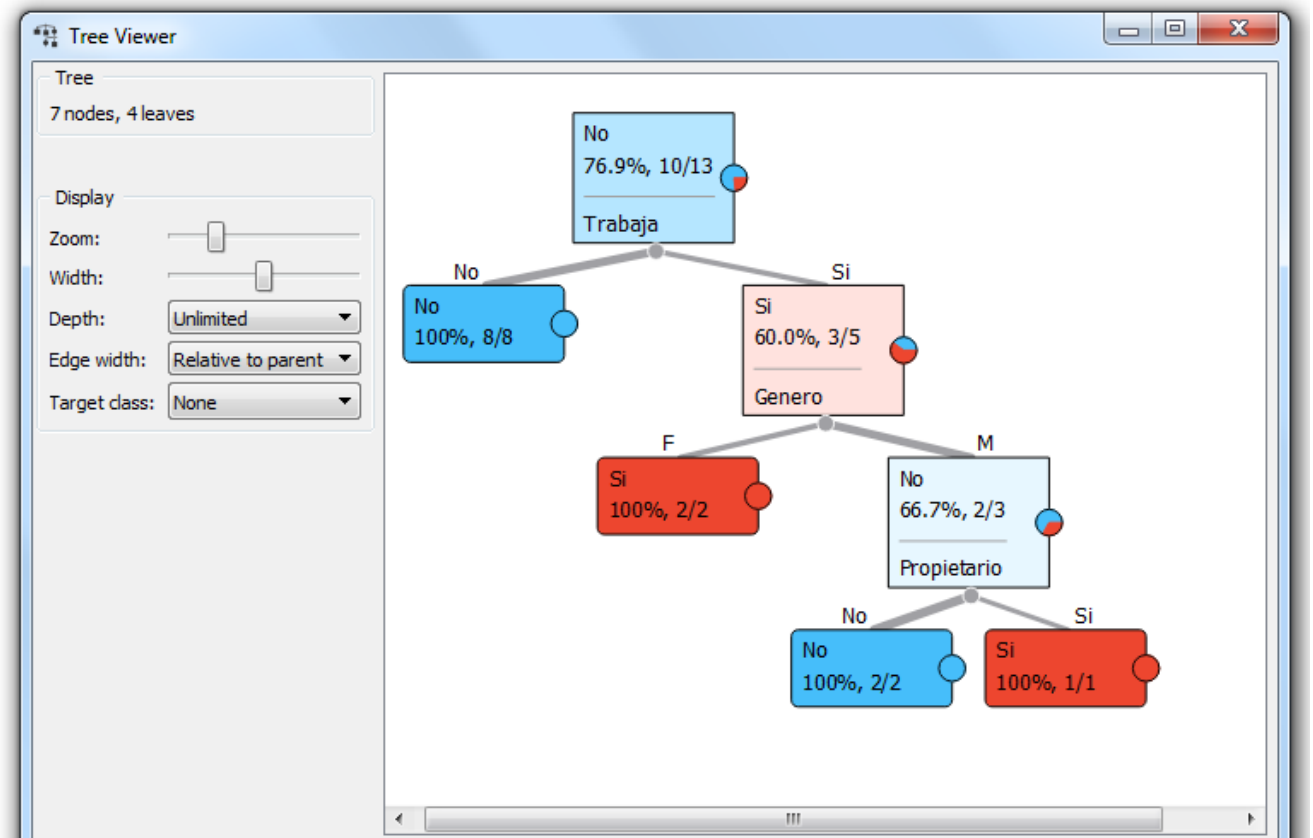
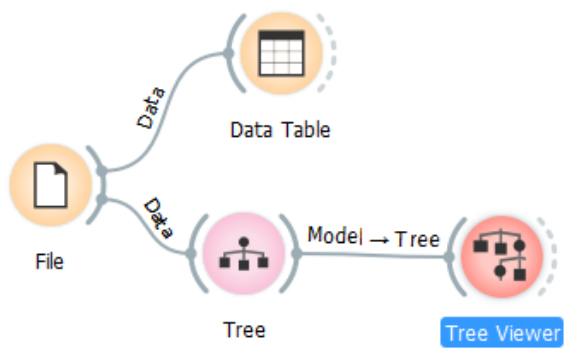


Aquí deberemos revisar algunos parámetros de nuestro árbol:

- **Induce Binary Tree:** cuando la variable por la que estamos particionando en un nodo tiene más de dos valores posibles se creará una rama para cada uno de estos valores; si estamos interesados en forzar a que en cada nodo siempre se generen sólo dos ramas debemos seleccionar esta opción. Por ejemplo si la variable puede asumir los valores: Bajo, Medio, Alto cuando particionemos por ella se creará una rama para cada caso, si deseamos que el árbol siempre sea **Binario** dos de estos valores se agruparán en una sola rama, por ejemplo una rama podría ser para Bajo/Medio y la otra rama para Alto. En nuestro caso la dejamos **sin seleccionar**.
- **Min number of instances in leaves:** Es la cantidad mínima de observaciones o instancias para que un nodo se considere como una hoja. Nosotros hasta ahora hemos considerado que para que un nodo sea considerado hoja debe tener todas las observaciones asignadas a una única clase (en nuestros ejemplos eran todos Si, o todos No), es decir la cantidad de observaciones que quedarían para asignar sería **cero**. Sin embargo, en el mundo real, muchas veces suele ser poco práctico generar nuevos nodos para distinguir entre sólo unas pocas observaciones, si esa es nuestra situación podemos indicar que cuando en un nodo queden por ejemplo 3 observaciones de distinta clase (algunos Si y otros No) que considere a ese nodo como hoja y no cree un nuevo nodo para distinguir los casos, lo cual agregaría más profundidad al árbol. Sirve para **podar** el árbol y no dejarlo que sea muy profundo.

Tree Viewer

Finalmente debemos visualizar el árbol resultado, para ello deberemos utilizar un widget del **panel de visualización**, el **Tree Viewer**



Recordemos las observaciones que tenemos en nuestros datos:

Out[5]:

	Tiene_Deuda	Genero	Trabaja	Propietario	Dar_Credito
0	Si	F	No	Si	No
1	Si	M	No	Si	No
2	No	F	Si	Si	Si
3	No	F	Si	No	Si
4	No	F	No	No	No
5	No	M	Si	No	No
6	No	M	No	Si	No
7	No	M	No	No	No
8	Si	F	No	No	No
9	Si	M	Si	No	No
10	Si	M	No	No	No
11	No	F	No	Si	No
12	No	M	Si	Si	Si

Tomemos por ejemplo la primera de ellas, la identificada con el número de orden 0

Out[6]:

```
Tiene_Deuda    Si
Genero          F
Trabaja        No
Propietario     Si
Dar_Credito    No
Name: 0, dtype: object
```

y apliquémosle el modelo de árbol que hemos construido, recorriendo el árbol desde el primer nodo.

Primero nos pregunta si trabaja, en nuestro caso la respuesta es No, y según el árbol debemos ir por la rama izquierda, que desemboca en una hoja y nos dice que el resultado pronosticado para Dar_Credito es No; lo cual coincide con el valor real para esta observación que era justamente Dar_Credito = No.

Repitamos el proceso con otra observación por ejemplo la que tiene número de orden 2, veamos los valores cargados en nuestros datos:

Repitamos la operación para otra observación, por ejemplo la identificada con el número 2

Out[7]:

```
Tiene_Deuda    No
Genero          F
Trabaja        Si
Propietario     Si
Dar_Credito     Si
Name: 2, dtype: object
```

Recorramos el árbol.

Primero nos pregunta si Trabaja, en esta observación la respuesta es Si, por lo cual debemos salir por la rama de la derecha del árbol.

En el nodo de la derecha nos pregunta por el Genero, en nuestro caso es F, por lo cual debemos seguir por la rama de la izquierda, y arribamos a un nodo que es una hoja, la cual nos pronostica que Dar_Credito es Si. Controlamos con la tabla donde tenemos nuestros datos y vemos que efectivamente Dar_Credito era Si.

Dos intentos y dos aciertos. Nuestro árbol funciona a la perfección.

Ejercicio:

Aplice el Árbol de Decisión a cada una de las observaciones que tenemos en los datos originales y observe si el resultado pronosticado para Dar_Credito por el árbol de decisión coincide con el de los datos originales.

Ud debe haber b que nuestro Árbol de Decisión clasifica la variable objetivo Dar_Credito correctamente en **todos** los casos, es decir con una **exactitud (accuracy)** del 100%, lo cual

extrañamente puede no ser tan bueno como parece

... pero analizaremos este tema más adelante.

Guarde el trabajo hecho con Orange3, el archivo tendrá una extensión .ows. Como nombre elija *clase_03_01_credito.ows*