

# Introducción a la ciencia de datos - PEC 2

## Presentación

*Evaluación continua (EC) de los contenidos lectivos correspondientes:*

- *Bloque 2: la ciencia de datos.*
  - o *Contenido: los roles, ámbitos y nombres de la Ciencia de datos (autor: Marçal Mora).*
  - o *Aspectos vinculados a:*
    - *¿Qué hace un científico de datos?*
    - *Técnicas de aprendizaje automático.*
    - *Open data y Open source.*
- *Bloque 3: ejemplos de proyectos de Ciencia de datos.*
  - o *Contenido: ejemplos de proyectos en el ámbito de la Ciencia de datos (autor: Marçal Mora).*
  - o *Aspectos vinculados a la visualización de los datos.*

## Objetivos y competencias

- *Comprender los conceptos utilizados habitualmente y relacionados con el contexto de la Ciencia de datos.*
- *Identificar las fases del ciclo de vida del dato.*
- *Entender la necesidad de poner en práctica otras competencias a la hora de trabajar poniéndonos en el papel de un científico de datos:*
  - o *Entender las fases del proceso ETL.*
  - o *Construir visualizaciones para el análisis de datos.*
  - o *Entender los distintos tipos de aprendizaje automático y metodologías de la Ciencia de datos.*

## Declaración de trabajo original (no plagio) del estudiante

Yo, **NombreEstudiante**, declaro que para realizar esta entrega... *(completa la frase con sus propias palabras)*

## Enunciado de la PEC

## PEC 2: Ejemplos prácticos de análisis

### **Criterios de evaluación generales de la PEC.**

- *Aportación de alguna referencia externa que complemente o sustente los razonamientos que se expongan.*
- *No omitir ninguno de los apartados de cada ejercicio.*
- *Respetar la extensión de palabras señalada en cada enunciado.*
- *Claridad en las respuestas y en los razonamientos.*
- *Capacidad de síntesis.*
- *Originalidad.*
- *Sólo se permite el uso de **herramientas de IA** para*
  - *Búsqueda de información*
  - *Mejora de la ortografía, la gramática y la claridad de un texto.*
  - *Traducción de un texto.*
  - *Cualquier otro uso no está permitido.*

### **Formato de entrega**

- *Las respuestas se entregarán en **formato PDF y nunca en paquete .ZIP o similar.***
- *Las respuestas se subirán a la caja de **Contenidos del aula correspondiente**, denominada “PEC2, Entrega estudiantes”.*
- *El documento **no debe incluir el enunciado**, solo las respuestas.*
- *El documento debe estar **estructurado** y el texto en un color que facilite la lectura (negro o azul oscuro).*
- *La **fecha máxima** para entregar las respuestas es el **12 de mayo de 2024, a las 23:59h**. No se corregirán las PEC que no cumplan este requisito, excepto en casos de fuerza mayor y debidamente justificados.*

## Pregunta 1: Periodismo de datos (20%)

### Enunciado

A lo largo del módulo hemos visto variedad de casos de uso (ejemplos de aplicación) de la ciencia de datos, con objetivos diversos. Una rama que está ganando creciente interés es el llamado "periodismo de datos", en el que el periodismo (o los medios en general) combinan los datos disponibles con atractivas (e, idealmente, explicativas) visualizaciones con un doble objetivo: explicar más y mejor, dejando incluso cierta interactividad al usuario. Aquí tenéis dos ejemplos:

- Tracking global data on electric vehicles (OurWorldinData.org, 2024) [consulta: 7 de Abril de 2024]
- Artificial Intelligence (OurWorldInData.org 2023) [consulta: 7 de abril de 2024]

Echad un vistazo a cada uno de ellos y escoged el que más os guste. Haced una lectura comprensiva (e interactiva) del artículo de vuestra elección y elaborad un resumen (300-500 palabras) que incluya:

- El contexto y objetivos del estudio.
- Los datos que se han usado.
- Los principales resultados/conclusiones.
- ¿Qué te ha llamado más la atención?
- ¿Qué te gustaría que se hubiera analizado y cómo lo harías?

### Criterios de evaluación

1. Se valorará que la respuesta incluya todos los puntos descritos en el guión.
2. Se valorará la interpretación y mejoras propuestas del caso de Ciencia de Datos existente.
3. Se valorará que se hayan examinado las visualizaciones e interactuado con ellas.
4. El resumen total tendrá entre **300 y 500** palabras. En caso contrario, se reducirá la puntuación máxima total a la mitad (**1 punto**).
5. Cada punto del guión se valorará con **0.4 puntos como máximo**.

## Pregunta 2: Machine Learning I (30%)

### Enunciado

A continuación, veremos los pasos iniciales necesarios para realizar un experimento de ciencia de datos, empezando por la instalación de una herramienta analítica, la obtención y transformación de los datos, y el análisis exploratorio de estos.

### 2.1 Configuración de la herramienta analítica

Para este experimento utilizaremos la herramienta analítica **Orange3**, la cual podemos descargar en <https://orangedatamining.com/download/>. Se trata de un **software libre** desarrollado en el laboratorio de Bioinformática de la Universidad de Ljubljana (Eslovenia) que permite analizar datos y crear modelos de aprendizaje automático de forma gratuita.

Una vez en la página de descargas, seleccionaremos nuestro sistema operativo y seguiremos las instrucciones de instalación (ver ilustración 1).

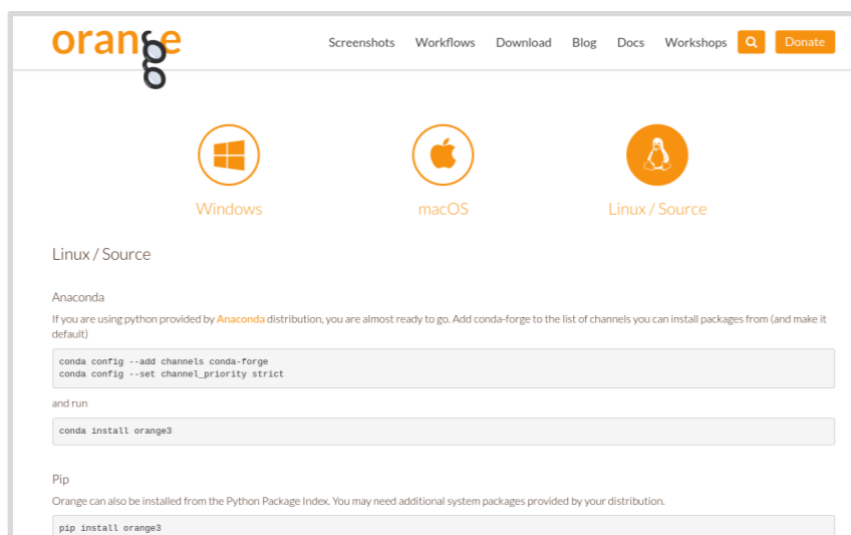


Ilustración 1

Si tenéis alguna versión anterior instalada, y tenéis algún problema con la instalación, os recomiendo que la desinstaléis antes de instalar la última versión de Orange3. La instalación no presenta ninguna dificultad, pero os dejamos unos links con un poco más de información por si fuera necesario:

- <https://www.youtube.com/watch?v=48XgkW4P6Lw>
- <https://biolab.github.io/install-orange/>

Una vez realizada la instalación, se os abrirá la ventana principal con un cuadro de diálogo (ver cuadro rojo de la ilustración 2).

A continuación, cerraremos esta ventana y guardaremos el proyecto con el nombre “**NombreApellido1**”, es decir, vuestro nombre y el primer apellido. Para ello, nos dirigiremos a la barra de navegación vertical (ver cuadro azul de la ilustración 2). Seleccionaremos “File” y en el desplegable seleccionaremos “Save as”.

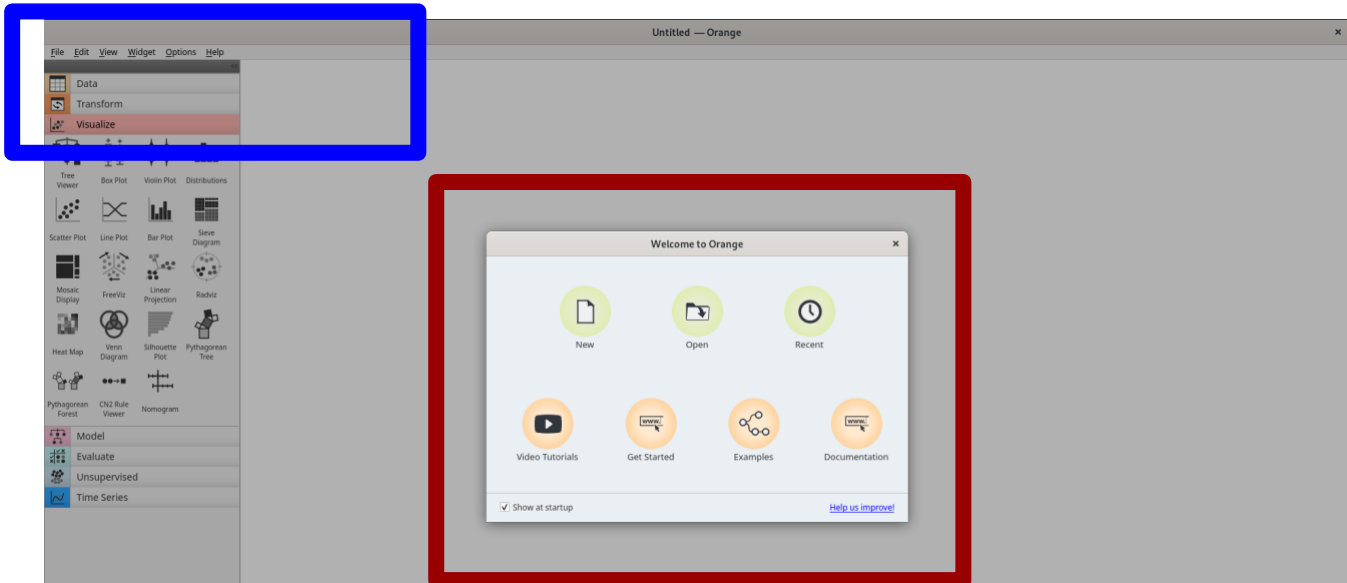


Ilustración 2

Se nos abrirá otro cuadro de diálogo donde podremos guardar el fichero (ver ilustración 3). Una vez hecho esto, aparecerá arriba de la pantalla el nombre del nuevo proyecto, indicando así que podemos empezar a trabajar en él.

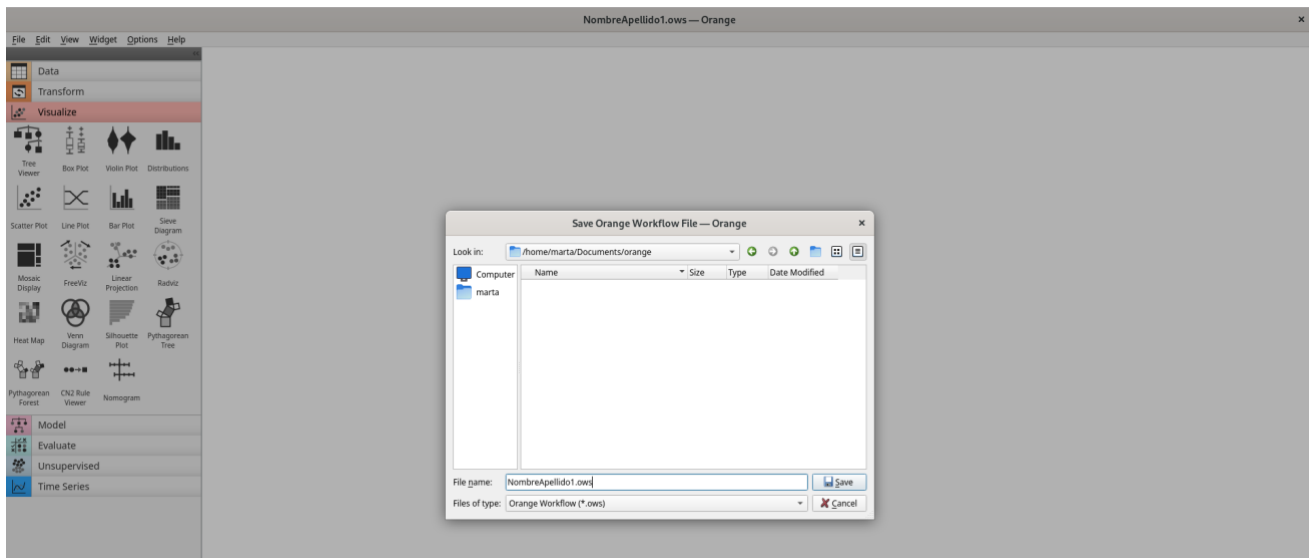


Ilustración 3

**Pregunta 2.1.** Haz una captura de pantalla similar a la imagen anterior, que muestre en el nombre del proyecto tu “NombreApellido1”.

**Importante:** Las próximas capturas de pantalla que incluyan el modelo deben mostrar este “NombreApellido1”

## 2.2 Obtención y análisis inicial de datos

En este experimento vamos a utilizar el dataset “**Diabetes Dataset For Beginners**”, el cual podemos encontrar en formato CSV en la siguiente competición de **Kaggle**, que es una comunidad de más de diez millones de científicos de datos.

[Diabetes Dataset For Beginners, Kaggle.com](https://www.kaggle.com/shantanu-dhakad/diabetes-dataset-for-beginners)

Para poder descargar el dataset, es necesario que os registréis previamente, con una cuenta de Google, o bien, con un correo electrónico.

Una vez dentro, podréis observar que esta competición pretende responder a una pregunta muy concreta. Analizad los datos que nos ofrece y responded:

**Pregunta 2.2.A** Describe brevemente el dataset y qué es cada atributo. ¿A qué pregunta(s) de investigación podría dar respuesta(s) este dataset?

Para descargar el dataset (diabetes.csv), tenemos que usar el botón “Download” (ver recuadro rojo de la ilustración 4) y hacer clic en él.

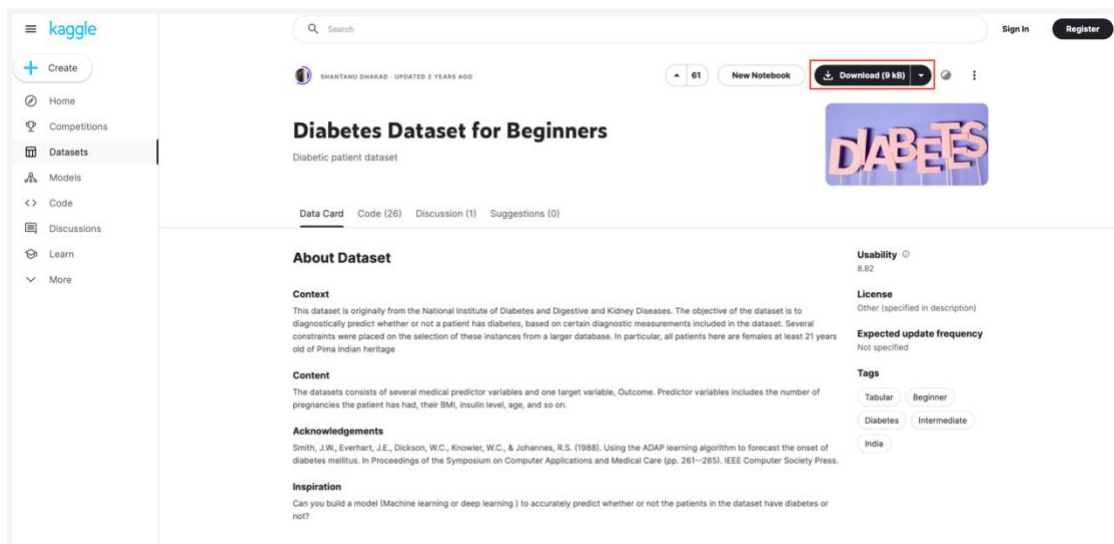


Ilustración 4

Una vez almacenado el fichero, volveremos a Orange3 para pre-procesar estos datos.

*[Las capturas de pantallas que observaremos a continuación están realizadas con la versión 3.36 para Mac OS, por lo que el aspecto de estas capturas, y puede que algún otro detalle*

(add-ons de la barra lateral), no coincide exactamente con los sistemas operativos que tenáis instalados o sus versiones.]

El siguiente paso será subir estos datos a Orange3. Para ello, iremos a nuestro proyecto “**NombreApellido1**”, y seleccionaremos en la barra de navegación de la izquierda, en el desplegable “**Data**” (el primero por arriba), el icono de “**CSV File Import**”.

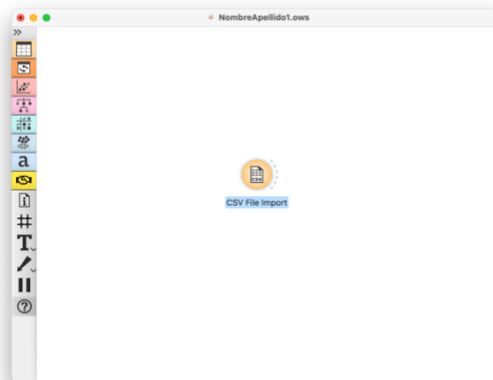


Ilustración 5

Al hacer clic en este (o cualquier) icono de la barra de navegación de la izquierda, este se nos colocará automáticamente en el “**canvas**” (espacio en blanco de Orange3), pero también lo podemos arrastrar y colocar en la posición que más nos guste. Otra opción, es hacer clic en el botón derecho del ratón, y se nos abrirá una lista de objetos, y una barra de búsqueda, para que podamos localizar o buscar el icono y desplegarlo en el canvas.

Para poder cargar los datos, haremos clic en este icono una vez esté en el canvas, y a continuación se nos abrirá una ventana como la de la ilustración 6.

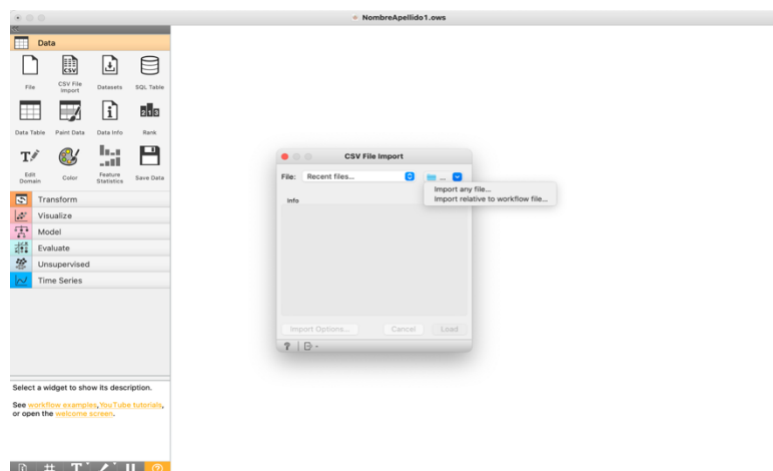


Ilustración 6

Seleccionamos el icono de la carpeta azul, y seleccionamos **“Import any file”**. A continuación, seleccionaremos el fichero **“diabetes.csv”** que tendremos en la carpeta en donde lo hemos descargado previamente. Una vez cargado, esta ventana nos muestra una tabla con la cantidad de filas, o instancias del dataset (769), el número de atributos (9) y meta atributos(0). En este punto, podemos obtener una visualización del dataset, haciendo clic en el botón **“Import Options...”** (ver ilustración 7). Haremos clic en el botón **“Ok”** y cerraremos la ventana del **“CSV File import”**.

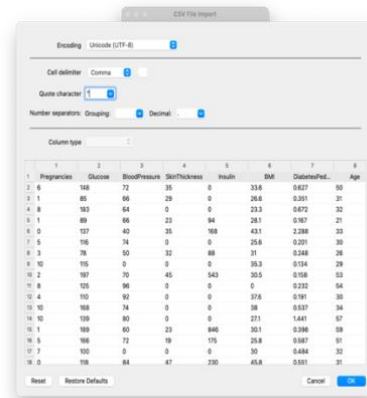


Ilustración 7

Otra manera de obtener esta información, especialmente cuando el dataset es muy grande, es conectar un icono **“data info”** a la salida del icono **“CSV Import File”**. La manera más rápida de hacer esto, la cual os recomiendo para desplegar objetos en el canvas, es la siguiente:

- Hacemos clic en la barra curva lateral del icono **“CSV Import File”** y sin soltar, arrastramos la línea que aparecerá en ese momento, y finalmente, soltamos el botón del ratón (ilustración 7.a).
- Esto abrirá el menú de objetos disponibles para conectar a la salida de nuestra fuente de datos.
- Escogemos el primer elemento de la lista, que es el icono **“Data Info”** (también se puede usar el buscador)



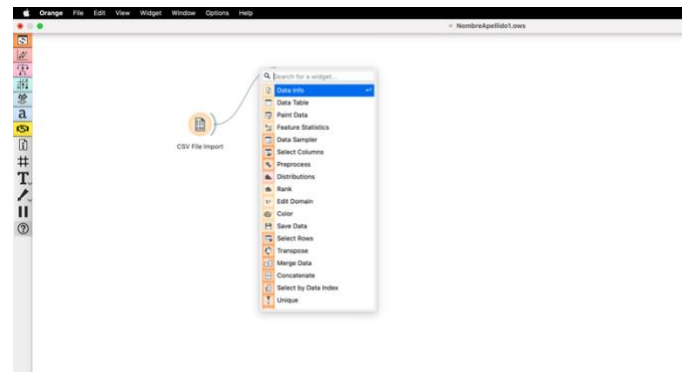


Ilustración 7.a

Una vez hecho esto, ya tendremos conectados ambos iconos, y si hacemos clic en el icono “**Data Info**”, veremos la información resumida de nuestro dataset.



Ilustración 7.b

Este mismo proceso se puede realizar escogiendo a priori los iconos, soltándolos en el canvas, y conectándolos entre ellos posteriormente.

Una vez hemos cargado los datos, vamos a analizarlos. Para ello, seleccionaremos el icono de “**Feature Statistics**”, y una vez nos aparezca en el canvas, conectaremos nuestra fuente de datos a este nuevo icono. Para ello, hacemos clic en el lateral del icono de la fuente de datos y arrastramos la línea hasta el icono de “**Feature Statistics**” (ver ilustración 8).

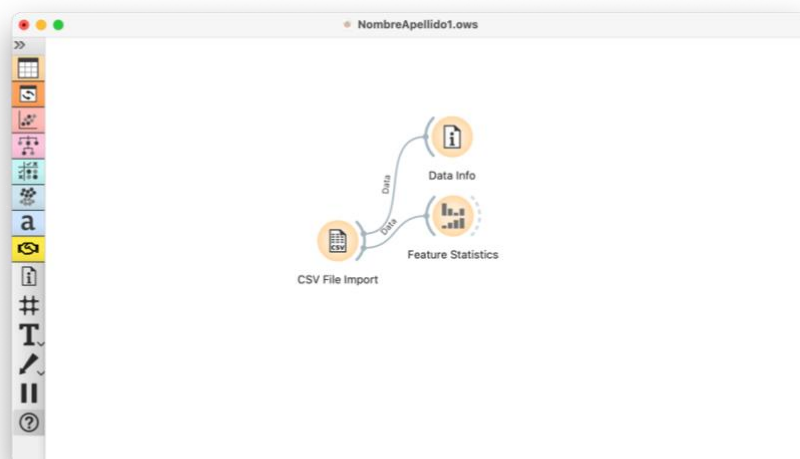


Ilustración 8

A continuación, haremos doble clic en el icono de “Feature Statistics” del Canvas, y se nos abrirá la siguiente ventana (ver ilustración 9).

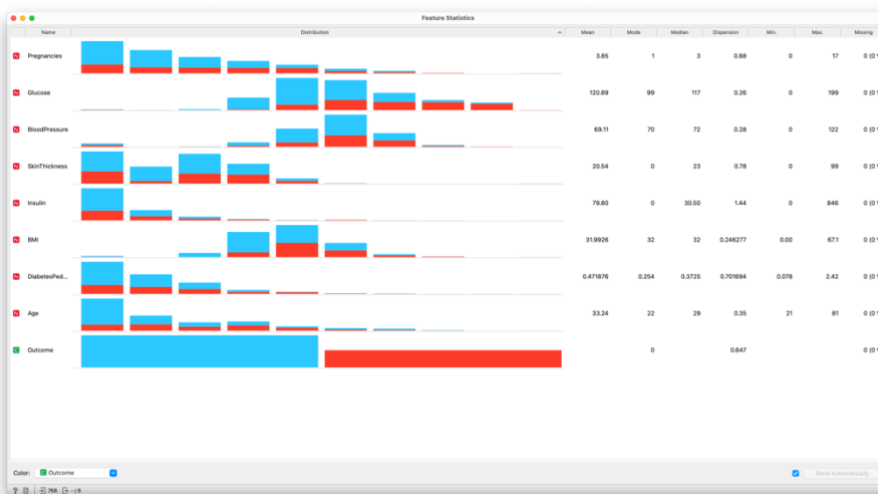


Ilustración 9

**Pregunta 2.2.B.** Adjunta dos capturas de pantalla, cómo las ilustraciones 8 y 9 y analiza las estadísticas de cada uno de los atributos de datos del dataset, el tipo de dato que contiene (simple, compuesto...), y si hay algún atributo que le falte valores.

**¡Importante!** Recuerda que la captura de pantalla de la ilustración 8 debe tener tu NombreApellido1 en el título del proyecto.

## 2.3. Transformación de los datos

Más adelante, vamos a entrenar un modelo de aprendizaje automático que nos **prediga la probabilidad de que un paciente pueda padecer la enfermedad**. En este tipo de entrenamientos, es mejor preparar los datos, ya que según el modelo a entrenar nos podría dar error o podría dar un resultado poco óptimo.

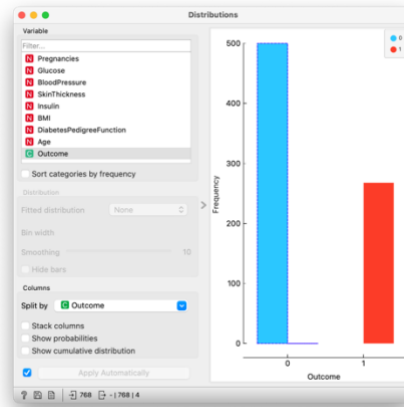
En este sentido, necesitamos que los datos no tengan sesgos estadísticos para los atributos que queremos analizar, en nuestro caso particular el atributo “**outcome**”, pues es el que indica si un paciente tiene la enfermedad.

En primer lugar. Vamos a analizar nuestra clase “**outcome**”, para esto vamos a conectar a nuestro origen de datos “**CSV Import File**” un icono “**Distributions**”, cómo podemos ver en la ilustración siguiente



Ilustración 10

A continuación, hacemos clic en el icono “**Distribution**”, y aparecerá una ventana como la de la ilustración 11, en donde automáticamente se ha seleccionado la clase “**outcome**” para la clasificación de las instancias (“Split by”). Seguidamente, entre la lista de variables disponibles del dataset escogemos la clase “outcome”, y podemos observar cómo hay 500 instancias con valor ‘0’ (65% de las instancias, representadas en color azul) correspondientes a pacientes que no tienen la enfermedad, y 268 instancias con valor ‘1’ (34,90% de las instancias, representadas en color en rojo) correspondientes a pacientes que tienen la enfermedad.



Il·lustració 11

Acabamos de detectar un motivo de **sesgo estadístico** (*bias* en inglés) en nuestro dataset para los valores de la clase “**outcome**”, ya que no tiene el mismo número de instancias para los dos valores, y cuando esto sucede decimos que el **dataset no está balanceado**. En caso de entrenar un modelo de *machine learning* con estos datos, los resultados no serían significativos, ya que el modelo predeciría mejor los casos negativos que los positivos de diabetes.

Para solucionar este problema de nuestro dataset no balanceado existen dos técnicas:

- **Over Sampling:** Consiste en igualar el número de instancias para los distintos valores de la clase, aumentando el dataset con instancias que tienen el valor con menos representación. En nuestro caso, tendríamos que aumentar el número de instancias con valor ‘1’ para la clase “outcome”, es decir, el número de casos positivos.
- **Under Sampling:** Consiste en igualar el número de instancias para los distintos valores de la clase, disminuyendo el número de instancias que tienen el valor con más representación. En nuestro caso, tendríamos que disminuir el número de instancias con valor ‘0’ para la clase “outcome”, es decir, el número de casos negativos.

En nuestro caso, vamos a ver la primera de estas técnicas, el **over sampling**. Si estuviéramos analizando un caso real, deberíamos buscar más datos e incorporarlos al dataset. Pero, por otro lado, en medicina y en otros dominios, a menudo se tienen datos no balanceados, pues son muchos más abundantes los casos negativos que los positivos para una determinada clase.

Para aumentar el número de instancias positivas, existen **técnicas sintéticas**, que crean nuevas instancias a partir de las existentes, y **técnicas de muestreo**, que seleccionan muestras aleatorias entre las ya existentes, y las duplican hasta conseguir el número deseado. Para este caso de estudio, y para simplificar, vamos a utilizar la técnica de muestreo.

En primer lugar, vamos a conectar un icono “**Select Rows**” a nuestro origen de datos “**CSV File Import**”, tal y como se muestra en la ilustración 11b. Con este nuevo objeto, vamos a seleccionar todas las instancias que tienen el valor ‘1’ para la clase “outcome”. Para esto, haremos clic en el objeto “**Select Rows**”, y en la nueva ventana seleccionaremos la clase “outcome”, la condición “is”, y el valor 1, tal y cómo podemos ver en la ilustración 11c.

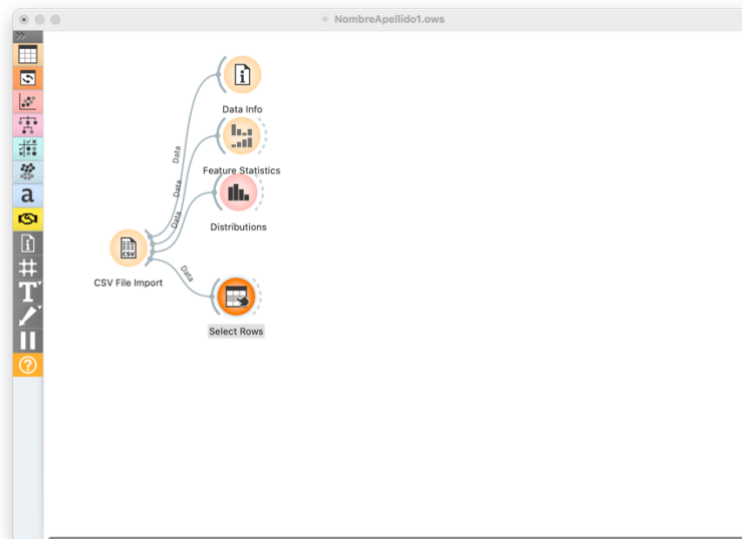


Ilustración 11b

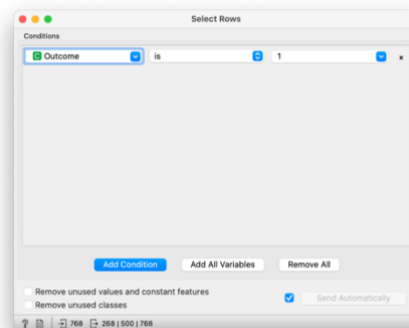
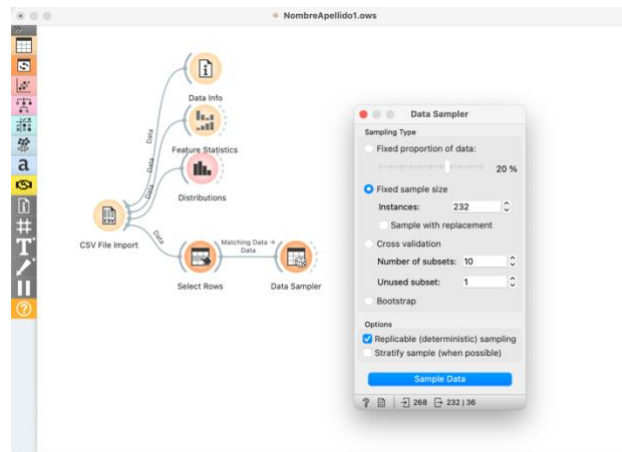


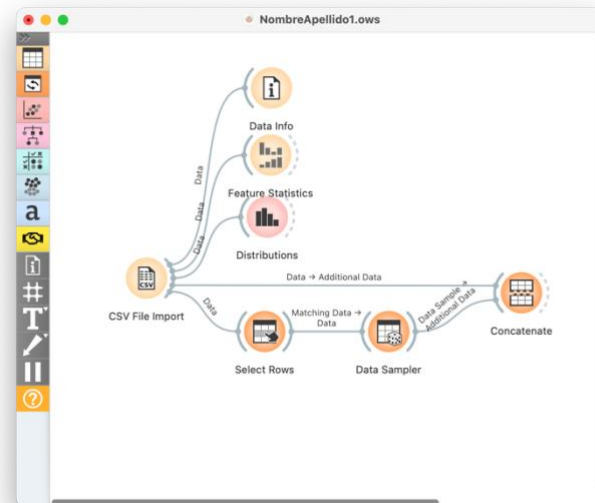
Ilustración 11c

A continuación insertaremos un objeto “**Data Sampler**” y lo conectaremos al objeto “**Select Rows**” anterior. Con este nuevo objeto, vamos a tomar una muestra aleatoria de instancias, tantas como sean necesarias para igualar la cantidad de instancias de la clase ‘0’ (500 instancias). Para ello haremos clic en el objeto “Data Sampler” y se abrirá una ventana en donde seleccionaremos la opción “**fixed sample size**”, e introduciremos el número de instancias que necesitamos para igualar la muestra, en nuestro caso ‘232’, ya que inicialmente tenemos ‘268’ instancias con valor ‘1’. Observad en la ilustración 11d, cómo configurar el objeto “Data Sampler”.



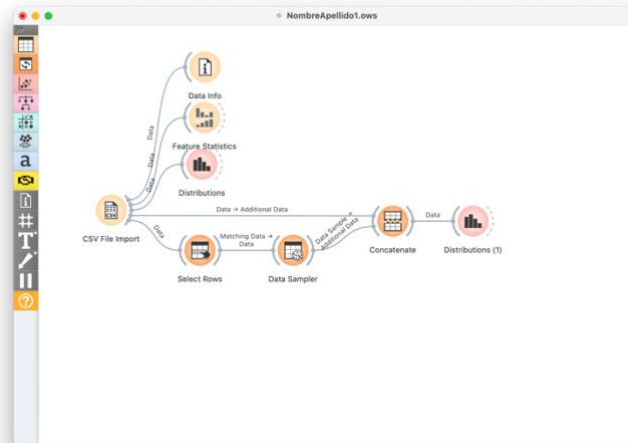
Il·lustració 11d

Con esto, ya tenemos las instancias necesarias de la clase “outcome” con valor ‘1’ y vamos a fusionarlo con nuestro dataset original. Para ello, vamos a colocar un objeto “**Concatenate**” y lo conectaremos, por un lado, a nuestro objeto “**Data Sampler**”, y, por otro lado, lo conectamos a nuestro origen de datos “**CSV File Import**”, tal y como se puede observar en la siguiente ilustración:

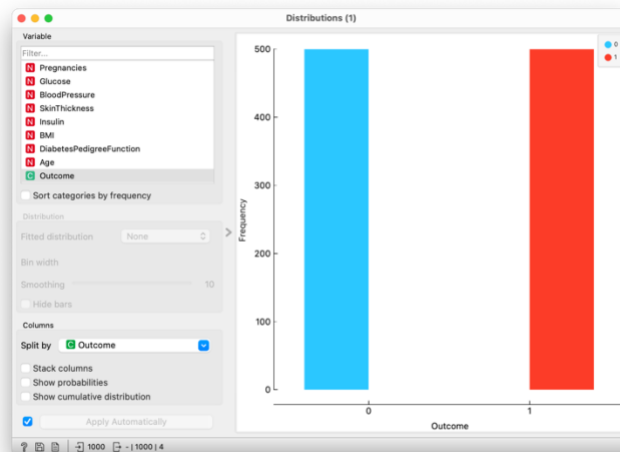


Il·lustració 11e

Para comprobar el resultado, conectaremos otro objeto “**Distribution**” al objeto “**Concatenate**”, y observaremos como ya tenemos el dataset balanceado (ilustraciones 11f y 11g).



Il·lustració 11f



Il·lustració 11g

**Pregunta 2.3.A.** Adjunta captures de pantalles donde se vea el proceso realizado para balancear el dataset, y que cómo mínimo, incluya las captures de pantalla de las ilustraciones: 11c, 11d, 11f y 11g.

**Importante:** Recuerda que tiene que verse el “NombreApellido1” del proyecto en las captures donde está el modelo desplegado.

A continuació, vam a comprovar que els canvis introduïts en els dades no tenen un impacte significatiu a nivell estadístic respecte al dataset inicial. Per això vam a utilitzar els dos objectes “**Distribution**” que tenim connectats al dataset original (“CSV file Import”) i al nou dataset generat (que tenim en el objecte “Concatenate”), per analitzar un mateix atribut en ambos datasets.

En primer lugar, aprovechando el objeto **“Distribution”** conectado al objeto **“Concatenate”** que acabamos de ver (ilustración 11g), seleccionaremos un atributo al azar, por ejemplo **“Glucose”**. Esto activará el cuadro inferior (distribution), que nos permitirá desplegar la lista **“Fitted distribution”** y seleccionar el valor **“normal”**. A continuación usaremos el deslizador de **“Bin width”** para llevarlo al valor mínimo y poder observar así el mayor número de barras. El resultado final lo podemos ver en la siguiente ilustración:

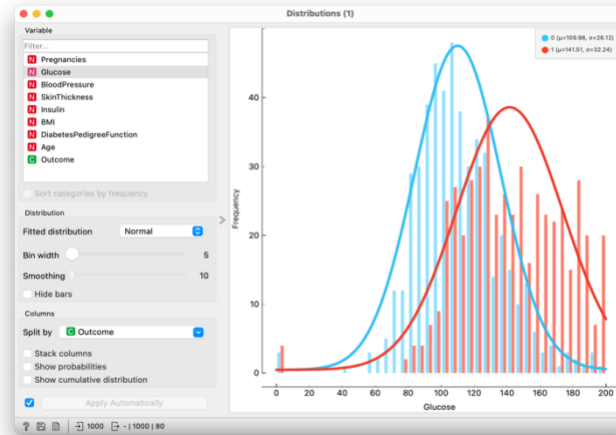
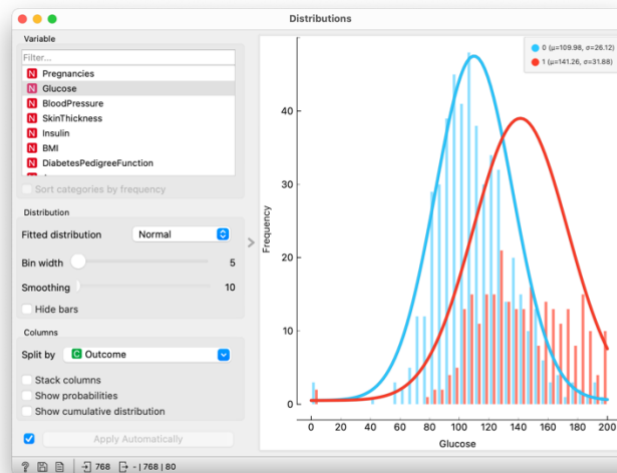


Ilustración 11h

En este caso, podemos observar que el valor ‘0’ de la clase **“outcome”** (color azul), correspondiente a los casos negativos en diabetes, sigue una distribución normal de media 109.98 y desviación estándar 26.12, mientras que el valor ‘1’ de la clase **“outcome”** (color rojo), correspondiente a los casos positivos en diabetes, sigue una distribución normal de media 141.51 y desviación estándar 32.24.

Realizamos el mismo análisis para el objeto **“Distribution”** conectado al dataset original **“CSV file Import”**, y observamos que a pesar de tener muestras diferentes para el valor de la clase ‘1’, obtenemos una distribución normal de media 141.26 y desviación estándar de 31.88, prácticamente la misma a la muestra original, por lo que el cambio que hemos realizado no ha tenido impacto significativo desde el punto de vista estadístico para este atributo. Observad que, para el valor ‘0’ la distribución es la misma en ambos casos, pues no hemos realizado ningún cambio en los datos.





Il·lustració 11i

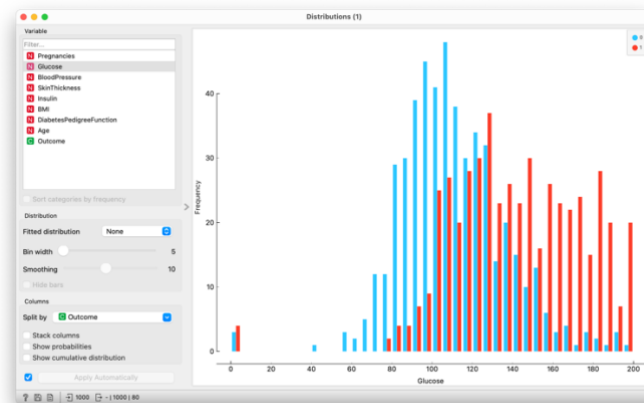
**Pregunta 2.3.B.** Escoge un atributo distinto del dataset y realiza el proceso anterior adjuntando las capturas de pantallas de las ilustraciones 11h y 11i, y analiza las distribuciones para el valor '1' de la clase "outcome" como en el ejemplo anterior.

## 2.4. Análisis exploratorio de los datos

A continuació, vamos a realizar una inspección visual del dataset, con el objetivo de detectar a simple vista qué atributos tienen mayor o menor relevancia al detectar la enfermedad.

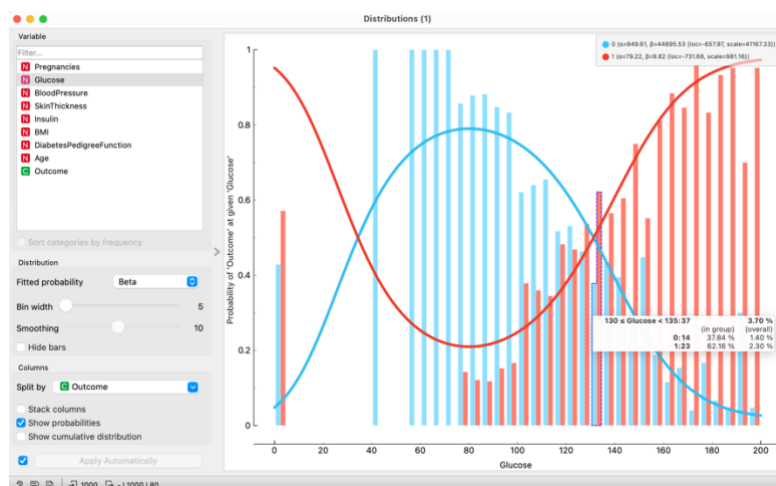
Para empezar con este análisis exploratorio de los datos (en inglés EDA, *exploratory data analysis*), abriremos de nuevo el objeto "**Distributions**" que tenemos conectado a nuestro dataset balanceado en el objeto "**Concatenate**" (ilustración 11f).

En este caso, vamos a seguir analizando el atributo "**Glucose**", y para ello lo seleccionaremos de la lista de atributos disponibles, nos aseguramos que tenemos la clase "outcome" seleccionada en la lista desplegable "**Split by**" y el valor de "**Bin Width**" al mínimo deslizando el selector hacia la izquierda, tal y como se puede ver en la imagen siguiente:



Il·lustració 12

Seguidamente, validamos el *check box* de la parte inferior izquierda de la pantalla llamado “**Show probabilities**”, y a continuación desplegamos la lista “**Fitted probability**” escogiendo una distribución de probabilidad “**Beta**”, cómo podéis ver en la imagen siguiente:



Il·lustració 12a

En la ilustración anterior, tenemos la probabilidad de tener o no tener la enfermedad en función de los distintos valores de glucosa. En este sentido, la curva azul indica la probabilidad de no tener la enfermedad, mientras que la curva roja indica la probabilidad de dar positivo en diabetes. Si os fijáis, cuando la curva de un determinado color está por encima de la del otro color, indica que la probabilidad del valor de clase correspondiente es mayor, mientras que cuando está por debajo la probabilidad es menor. Por otro lado, si dejáis el ratón encima de los valores del histograma, aparecerá un cuadro de texto (como el de la ilustración 12a).

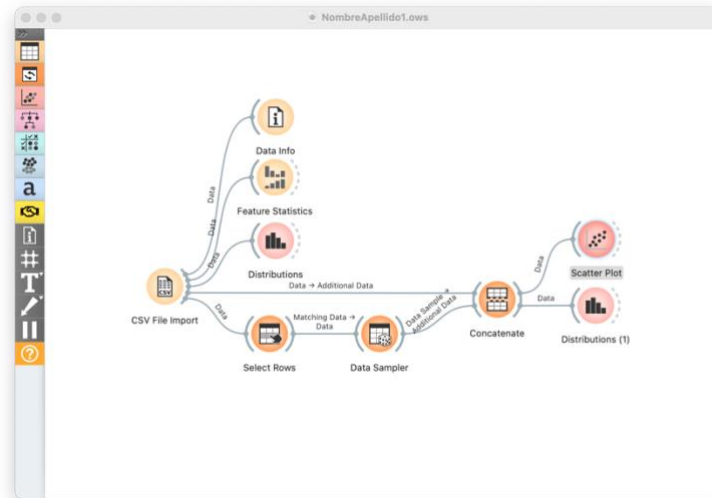
A continuación, analizando la ilustración, observamos que:

- A **valores extremadamente bajos de glucosa**, la probabilidad de tener la enfermedad es ligeramente superior a no tenerla (curva roja por encima de la azul). Fijaros en que los valores del histograma indican un 40% de probabilidad de dar negativo respecto a un 60% de probabilidad de dar positivo. Por otro lado, será necesario analizar si estos valores son correctos o no, pues están muy distanciados del resto de los valores de glucosa (outliers).
- A **medida que la glucosa va subiendo** hasta antes de alcanzar el valor de 80, la probabilidad de no tener la enfermedad es del 100%.
- A partir de **valores de glucosa de 80**, la probabilidad de no tener la enfermedad es mayor a tenerla, pero esta probabilidad empieza a decrecer a medida que aumentan los niveles de glucosa.
- Al llegar a **valores de glucosa comprendidos entre 130 y 137** (ver recuadro de la ilustración 12a), se produce un punto de inflexión, pues las gráficas se cortan indicando un cambio de probabilidad: 38% de dar negativo respecto a un 62% de dar positivo.
- A partir del **punto de inflexión anterior**, la probabilidad de tener la enfermedad aumenta claramente, siendo casi del 100% con valores de glucosa cercanos a 200.

**Pregunta 2.4.A.** Escoge un atributo de tu elección y realiza un análisis cómo el que acabamos de ver, aportando la captura de pantalla que muestre el atributo elegido y que refleje los puntos del análisis que se comentan.

Aparte de analizar las distribuciones de los parámetros de un dataset, **Orange3** permite obtener visualizaciones más elaboradas que permiten analizar el peso de los atributos de manera combinada en función de una determinada clase. A continuación, vamos a crear una visualización que nos permite analizar simultáneamente varios atributos.

Para esto, seleccionaremos la barra gris lateral de nuestro icono “**Concatenate**” del canvas para conectar (desde la ventana que aparecerá clicando y arrastrando la conexión) un objeto “**Scatter Plot**”. El resultado será como el de la ilustración 13.

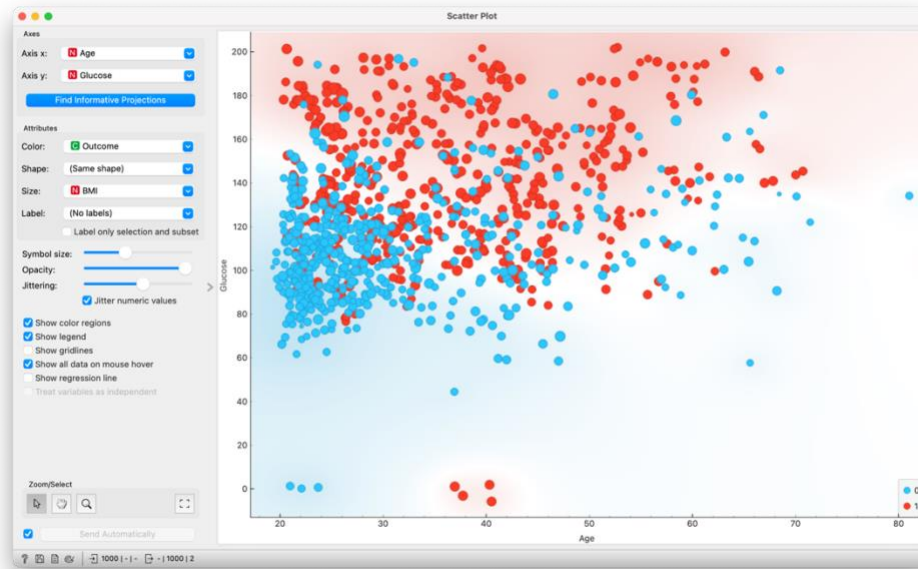


Il·lustració 13

A continuació obrim el objecte “Scatter Plot” haciendo clic en el icono, veremos una pantalla como la de la ilustración 14 en donde haremos lo siguiente:

- En el recuadro ‘Axis’ de la parte superior izquierda, escogeremos:
  - El atributo ‘**Age**’ en el eje ‘X’
  - El atributo ‘**Glucose**’ en el eje ‘Y’.
- En el recuadro ‘Attributes’, configuraremos:
  - El parámetro ‘**Color**’ con nuestra clase “outcome”
  - El parámetro ‘**size**’ con el atributo ‘BMI’
- Seleccionaremos los siguientes *check box*
  - ‘**Show color regions**’: nos permitirá colorear el fondo con el color del valor predominante de la clase para cada sección del gráfico.
  - ‘**Show all data on mouse hover**’: nos permitirá ver toda la información de una determinada instancia al dejar el ratón encima del correspondiente punto del gráfico.

El resultado final será como el de la siguiente ilustración:



Il·lustració 14

Observad cómo el fondo del gráfico, coloreado en azul y en rojo, nos da una idea de por donde tenemos más probabilidad de encontrar los valores positivos y negativos, en este sentido, podemos observar que la mayoría de los individuos jóvenes con niveles bajos de glucosa son negativos, pero a medida que aumenta la edad hacia los 50, los umbrales de glucosa bajan para los casos positivos, mientras que a partir de los 50 aproximadamente, los umbrales de glucosa suben para los casos positivos (forma de 'V' del fondo rojo).

Vemos también que, aparte del caso general, existen individuos (positivos y negativos) que se encuentran en una zona de color que no es la que le corresponde, y que, por tanto, no cumplen con la probabilidad prevista; observamos individuos con niveles muy altos de glucosa de diferentes edades que son negativos (punto azul) que están en la zona de probabilidad positiva (fondo rojo), y, por el contrario, individuos con niveles bajos de glucosa de diferentes edades que son positivos (punto rojo) que se encuentran en la zona de probabilidad negativa (fondo azul). Para analizar estos casos, podemos dejar el puntero del ratón encima de una instancia concreta y analizar el resto de los atributos.

Finalmente, en los extremos inferiores y derecho del gráfico, observamos a simple vista que hay instancias muy distanciadas del resto estadísticamente hablando, que en nuestro caso son:

- Las ocho instancias del extremo inferior del gráfico (3 negativas y 4 positivas) con valores anómalos de glucosa a '0'.
- La instancia azul, del extremo derecho del gráfico, que equivale a un individuo de más de 80 años.

Cómo podemos ver, esta exploración visual nos permite detectar estos valores atípicos (u **outliers** estadísticos), que pueden originar, en algunos casos, un mal comportamiento de nuestros modelos de *machine learning*, pues introducen sesgos estadísticos.

Aparte de esta visualización por defecto, la herramienta nos permite crear de manera automática un conjunto de visualizaciones con peso estadístico significativo. Para ello, utilizaremos el botón “**Find Informative Projections**”, que tenemos en la parte superior izquierda de la ilustración 14, y nos aparecerá una pantalla “**Score Plots**” como la de la ilustración 14b, en donde solamente tendremos que hacer clic en el botón “**start**” y obtendremos una lista de visualizaciones ordenadas de más a menos significativas estadísticamente (ilustración 14c).

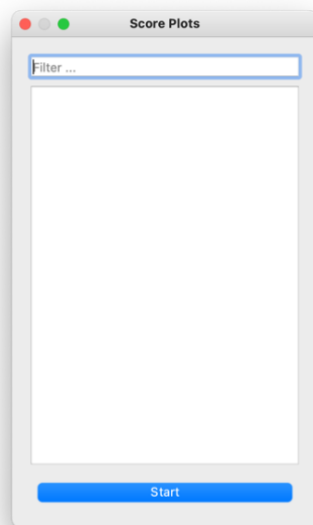


Ilustración 14b

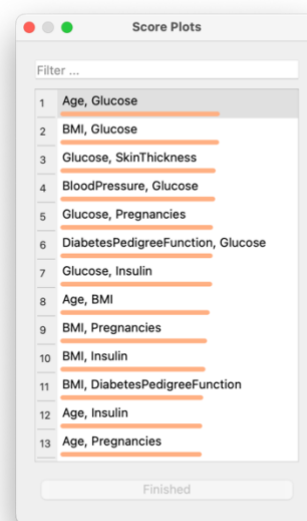


Ilustración 14c

Fijaros que cada elemento de la lista es un gráfico compuesto por los dos atributos indicados, y la barra naranja inferior, refleja la relevancia estadística de la combinación de ambos atributos para predecir los distintos valores de la clase (**outcome**). En este caso, observamos que el primer gráfico de la lista relaciona los atributos **Age** y **Glucose**, que es justamente el gráfico que hemos analizado anteriormente.

**Pregunta 2.4.B.** Realiza el proceso anterior, mostrando la captura de pantalla del proceso realizado (ilustración 13), que muestre tu nombre y primer apellido, escogiendo esta vez un gráfico de tu elección de la lista de visualizaciones (ilustración 14c), capturando la pantalla correspondiente (ilustración 14), analizándolo debidamente, y señalando algún elemento con comportamiento atípico.

A continuación, vamos a realizar el tratamiento de los outliers estadísticos detectados en el caso anterior por lo que, independientemente del gráfico que hayáis escogido en la pregunta anterior, vamos a trabajar con el gráfico analizado en este ejercicio (ilustración 14) y que podéis generar de manera automática con el botón “**Find Informative Projections**”, que tenemos en la parte superior izquierda de la ventana de nuestro objeto “**Scatter plot**” de la ilustración 14.

En primer lugar, vamos a clicar, arrastrar y soltar desde la barra gris de conexiones de nuestro objeto “**concatenate**”, y desde la ventana emergente seleccionaremos un objeto “**outliers**”, siendo el resultado el de la siguiente imagen:

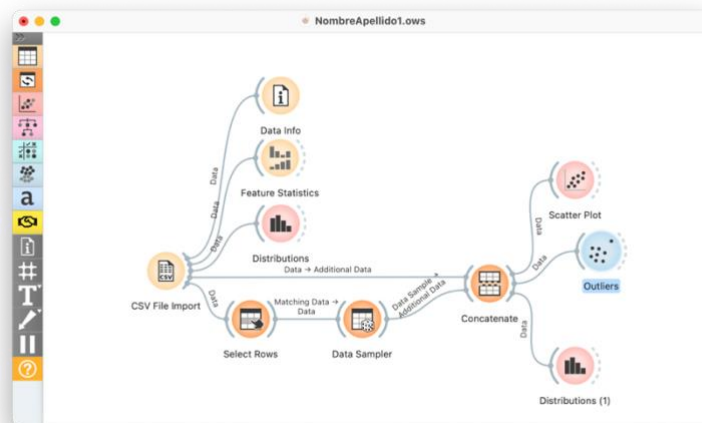
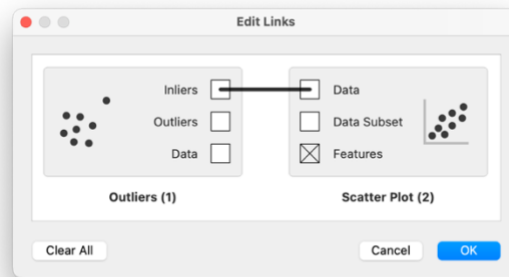


Ilustración 15

A continuación, conectaremos al nuevo objeto “**outliers**” a un nuevo objeto “**scatter plot**”, lo que hará que nos aparezca una pantalla emergente como la de la ilustración 15a de la siguiente página.

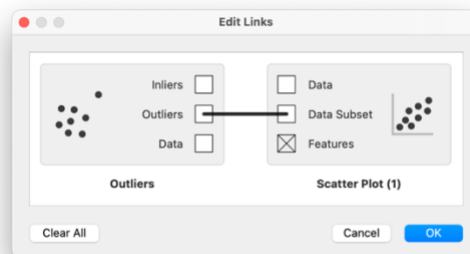
En primer lugar, vamos a eliminar la conexión “inliers-Data” que vemos en la imagen, apuntando y haciendo clic en la línea de unión.





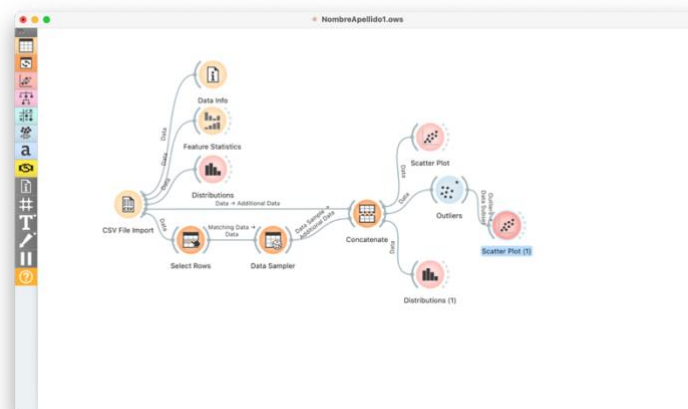
Il·lustració 15a

A continuació, presionarem en la casilla “**outliers**”, y sin soltar, conectaremos la nueva línea con la casilla “**data subset**”, siendo el resultado el de la siguiente imagen:



Il·lustració 15b

Una vez realizada la conexión, validamos con el botón ‘ok’, obteniendo como resultado una pantalla como la de la ilustración 16 de la siguiente página. Fijaros cómo por defecto, el nombre de la conexión entre los objetos “**outliers**” y “**scatter plot**” se llama “**outliers-data subset**”, lo cual es de utilidad para saber el tipo de conexión que hemos realizado, y si hacemos clic en la línea de la conexión, aparecerá de nuevo la ventana de configuración (ilustraciones 15a y 15b).



Il·lustració 16



Finalmente, para terminar la configuración enlazaremos nuestro objeto “**concatenate**” con el nuevo objeto “**scatter plot**”, siendo el resultado el de la siguiente imagen:

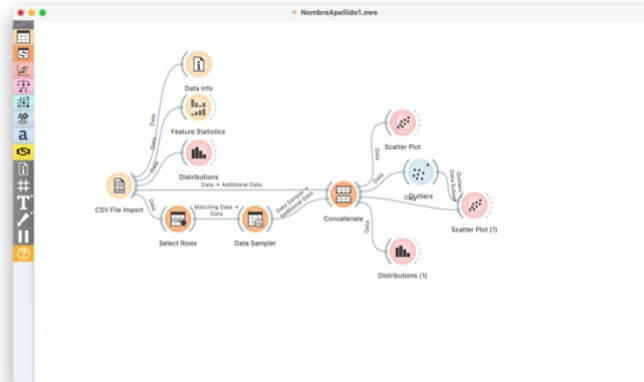


Ilustración 17

A continuación, vamos a abrir la ventana del objeto “**scatter plot**” y del objeto “**outliers**” y disponerlas una junto a la otra como en la ilustración 18. Acto seguido, vamos a configurar el objeto “**outlier**”, e iremos viendo cómo esto se refleja en el gráfico de puntos del objeto “**scatter plot**”, en donde podemos ver dos subconjuntos de datos, las instancias que son “**outlier**”, coloreadas en el color correspondiente al valor de la clase, y los que no se consideran outliers, con el interior del punto sin colorear.

Los parámetros que tenemos que configurar del objeto “**outliers**” son:

- **Method:** “Local Outlier Factor”
- **Contamination:** 4%
- **Metric:** es la distancia que vamos a usar para definir los *outliers*, en este caso, la **distancia coseno**.

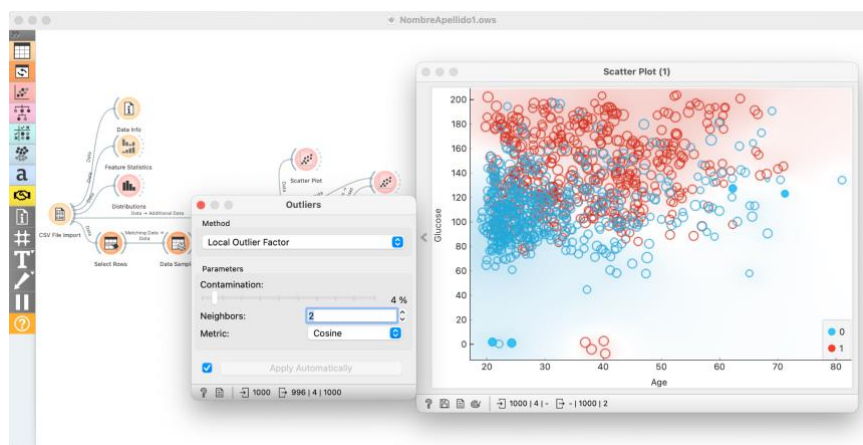


Ilustración 18

A continuación iremos aumentando manualmente de uno en uno el valor del parámetro “**Neighbors**”, y observaremos cómo los *outliers* detectados a cada nuevo valor del parámetro se van coloreando. Repetiremos el proceso hasta que se detecten las instancias que queremos considerar como *outliers*, en nuestro caso las instancias detectadas en la pregunta anterior. En este sentido, cuando llegamos al valor ‘10’ para el parámetro ‘neighbors’ se colorean las siete instancias con valores ‘0’ de glucosa (extremo inferior izquierdo), y la instancia con valor extremo para el atributo edad. Podemos ver el resultado final en la imagen siguiente:



Ilustración 19

Observad que, además de las instancias que detectamos visualmente en el apartado anterior, se han detectado otros *outliers*, en total 35 instancias, cómo podéis observar en la barra de estado inferior de la ventana ‘outliers’ de la ilustración 19.

Finalmente, conectaremos al objeto “**outliers**” un objeto “**Data Table**”, como en la siguiente ilustración

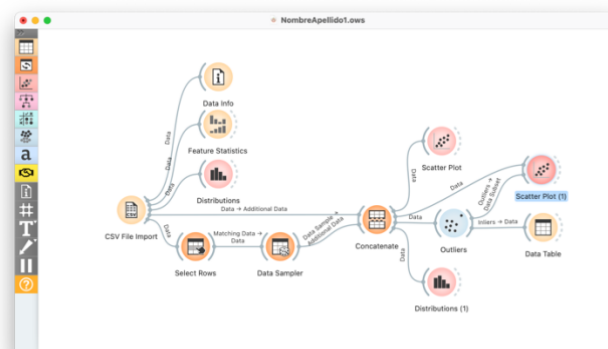


Ilustración 20

Haciendo clic en el nuevo objeto “**data table**” podremos observar que tenemos las 965 instancias de nuestro dataset (en el recuadro superior izquierdo) sin los 35 *outliers* eliminados.

Ilustración 21

Ahora que ya tenemos el dataset balanceado y sin *outliers* podemos implementar modelos de *machine learning*.

**Pregunta 2.4.C.** Realiza el proceso anterior, adjuntando las capturas de las pantallas similares a las ilustraciones 17,19,20 y 21 con tus explicaciones del proceso.

**¡Importante!** Recuerda mostrar tu NombreApellido1 en todas las capturas en las que aparezca el modelo.

## Criterios de evaluación

- La pregunta 1 se valorará con **0.5 puntos** como máximo
- La pregunta 2 se valorará con **0.5 puntos** como máximo, **0.25** por apartado
- Las preguntas 3.A y 3.B se valorarán con **0.5 puntos** como máximo cada una, con un total de **1 punto como máximo**.
- Las preguntas 4.A y 4.C se valorarán con **0.25 puntos** como máximo cada una, y la pregunta 4.B con **0.5 puntos** como máximo, con un total de **1 punto como máximo**.
- Todas las capturas que muestren el modelo implementado deben mostrar el nombre del proyecto “**NombreApellido1**”. Las que no lo tengan no serán valoradas.
- Todas las capturas de pantalla se tendrán que argumentar, tanto el contenido como el proceso realizado para su obtención. En caso contrario, se valorará **la mitad de la pregunta** como máximo.

- El **tamaño de las capturas** de pantalla ha de facilitar su análisis, y en caso de no poderse analizar, no se valorarán.

## Pregunta 3: Machine Learning II (30%)

### Enunciado

Ahora que ya conocemos los datos, los hemos analizado, y los hemos tratado, vamos a realizar experimentos más significativos desde el punto de vista estadístico.

Siguiendo con el ejemplo anterior, vamos a construir un **modelo supervisado** (que significa que conocemos a priori las etiquetas que nos permiten clasificar las instancias) que **nos permita predecir la probabilidad de que un paciente tenga diabetes**. Por tanto, lo que haremos es crear un **modelo de aprendizaje automático** que nos indique cuál es la probabilidad de que un determinado paciente dé positivo o negativo en función de los valores de los atributos del dataset que ya conocemos.

Cuando se construye un modelo por primera vez, es necesario hacer desarrollos en paralelo con distintos métodos y algoritmos para comparar y analizar los resultados obtenidos. Para este caso particular utilizaremos varias técnicas; la **regresión logística**, la **red neuronal**, y el algoritmo **Naïve Bayes**, para la construcción de nuestro modelo, y posteriormente, analizaremos el comportamiento en cada caso particular

Antes de empezar a aplicar esta técnica debemos preparar los datos.

### 3.1 Preparación de los datos

Al tratarse de una **técnica supervisada**, inicialmente es muy importante entrenar el algoritmo, esto significa que antes de realizar una clasificación, debemos darle un conjunto de **datos de entrenamiento** correctamente clasificados para que el algoritmo implemente el modelo óptimo para la clasificación. Por otra parte, debemos poder verificar el modelo de alguna manera, por lo que el algoritmo necesita un conjunto de **datos de prueba**.

En este punto es importante destacar que, en nuestro caso, disponemos de un **único dataset**, por lo que tendremos que subdividirlo en dos partes, un conjunto de datos de entrenamiento (train), y un conjunto de datos de prueba (test):

- **Train:** El nombre ya nos indica que será el conjunto de datos para el entrenamiento. Observad que tenemos las “etiquetas” necesarias para implementar métodos supervisados, ya que disponemos de la clase **outcome**.
- **Test:** Estos datos nos servirán para validar el entrenamiento anteriormente comentado. En base a esto, necesitamos crear este conjunto de datos de prueba,

para poder evaluar el comportamiento del modelo, una vez entrenado con los datos del conjunto de **train**.

Para esto podemos dividir el dataset original en dos partes:

- un **conjunto de entrenamiento o train** con el 80% de los datos del dataset.
- un **conjunto de prueba o test** con el 20% de los datos del dataset, para evaluar el comportamiento del modelo.

Antes de hacer esto, tenemos que seleccionar el atributo sobre el que vamos a entrenar a nuestro modelo, en este caso la clase **“outcome”**. Para esto, conectaremos al objeto **“outliers”** un objeto **“Select columns”**, tal y como podemos ver en la siguiente imagen:

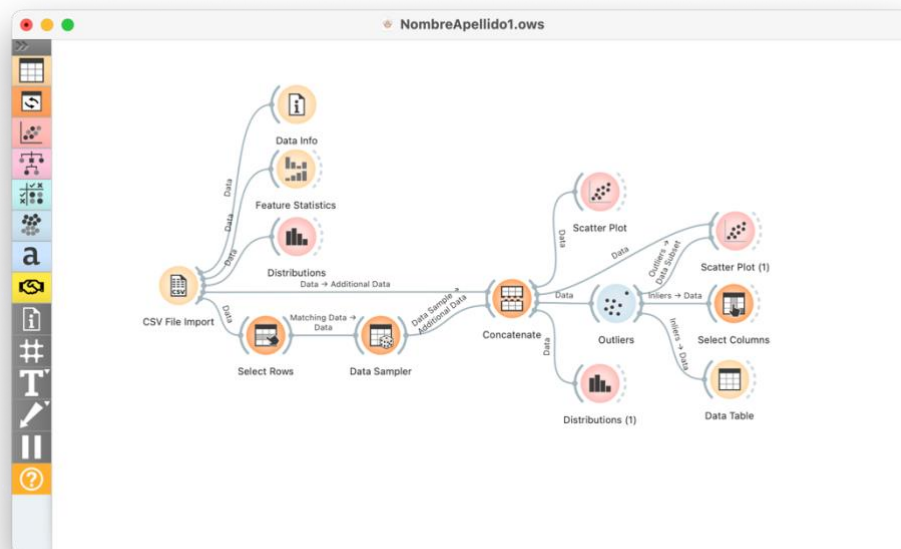
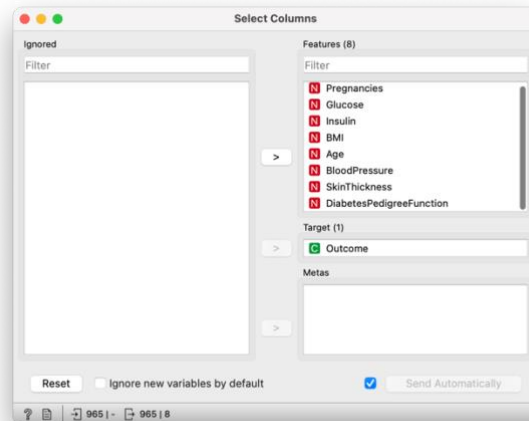


Ilustración 22

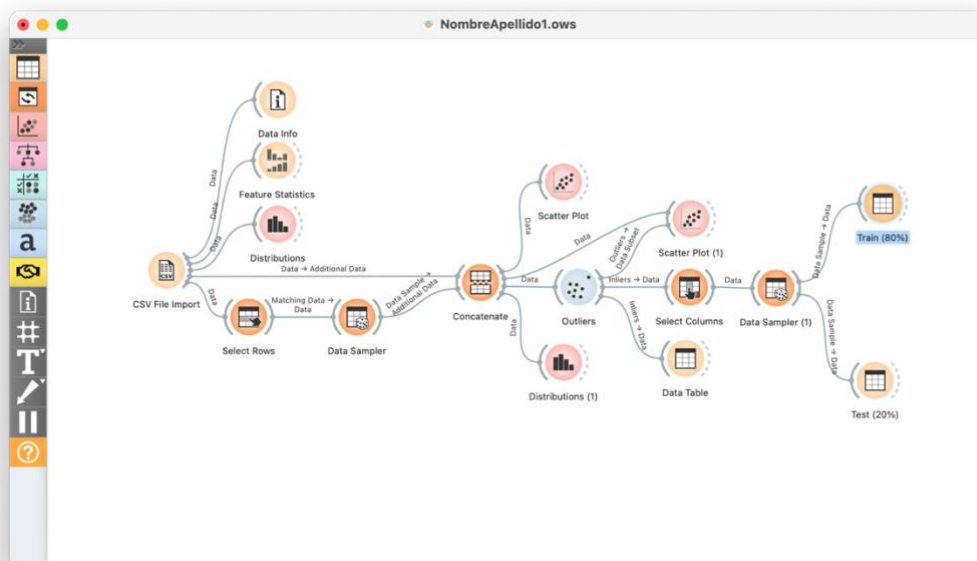
A continuación, haremos clic en el nuevo objeto **“select columns”** y seleccionaremos la clase **“outcome”** en la ventana **‘Target’** usando las flechas **‘>’** y **‘<’** para mover los atributos entre ventanas. El resultado final será el de la ilustración 22.

Acto seguido, ya podemos proceder a dividir el dataset original en los dos subconjuntos. En primer lugar, y tal como hicimos para balancear el dataset, conectaremos un objeto **“Data Sampler”** al objeto **“Select columns”**.



*Ilustración 22*

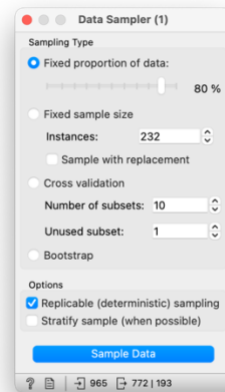
A continuación, conectaremos dos objetos “**Data Table**” al objeto “**Data sampler**”. Una vez hecho esto, nombraremos los objetos de datos usando la opción *rename* (usando botón derecho del ratón o haciendo clic en el nombre del objeto seleccionado), para cambiar los nombres a “*Train (80%)*” y “*Test (20%)*”, siendo el resultado final igual al de la ilustración 23.



*Ilustración 23*

Una vez tenemos esto ya preparado, vamos a construir los conjuntos de datos propiamente. Para ello haremos clic en el icono “**Data Sampler**”, y se abrirá una ventana de configuración como la de la ilustración 23a, y que deberemos configurar de la misma manera,

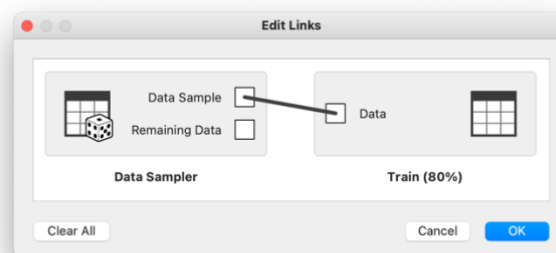




Il·lustració 23a

Es importante asegurarse de dejar la opción **sampling type** a fija con un valor del 80%.

Una vez hecho esto, haremos clic en el arco que une el icono “**Data Sample**” con el icono “**Data Table**” que hemos renombrado cómo “**Train (80%)**” y se abrirá una ventana como la siguiente.

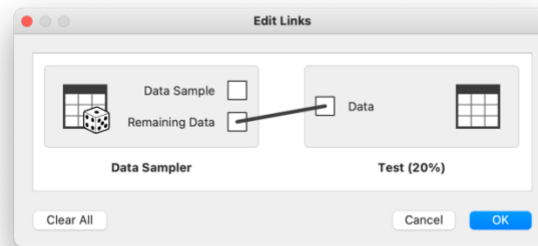


Il·lustració 23b

En esta ventana nos aseguraremos de que el flujo es el mismo que observamos en la ilustración 23b, es decir, desde “**Data Sample**” a “**Data**” entre ambos objetos.

Hacemos el mismo proceso para el arco que une el icono “**Data Sample**” con el icono “**Data Table**” que hemos renombrado cómo “**Test (20%)**” y se abrirá una ventana como la siguiente.

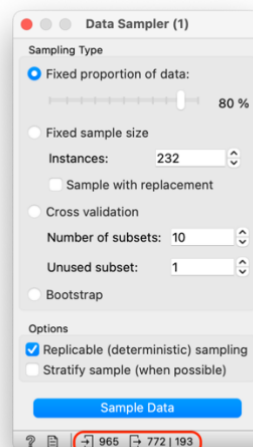




Il·lustració 23c

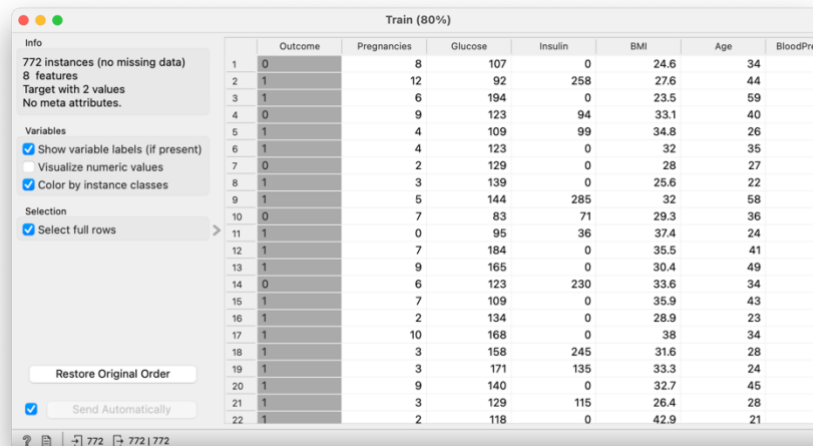
En esta ventana nos aseguraremos de que el flujo es el mismo que observamos en la ilustración 23c, es decir, desde **“Remaining Data”** a **“Data”** entre ambos objetos. En caso de que tengamos que modificar el flujo, basta con clicar en una conexión existente para eliminarla, y basta con seleccionar el origen nuevo y el destino para dar de alta un nuevo flujo de datos. Recordad utilizar el botón **“ok”** para validar los cambios.

Una vez realizado esto, volveremos a la configuración del icono **“Data sampler”** (ilustración 23a) y usaremos el botón **“Sample Data”** para refrescar y validar la carga de los datos en base a las configuraciones que hemos realizado. Fijaros cómo una vez hecho esto, observamos ya el reparto de datos (772 y 193 respectivamente) en la barra de estado de la ventana de configuración (recuadro rojo de la ilustración 23c)



Il·lustració 23c

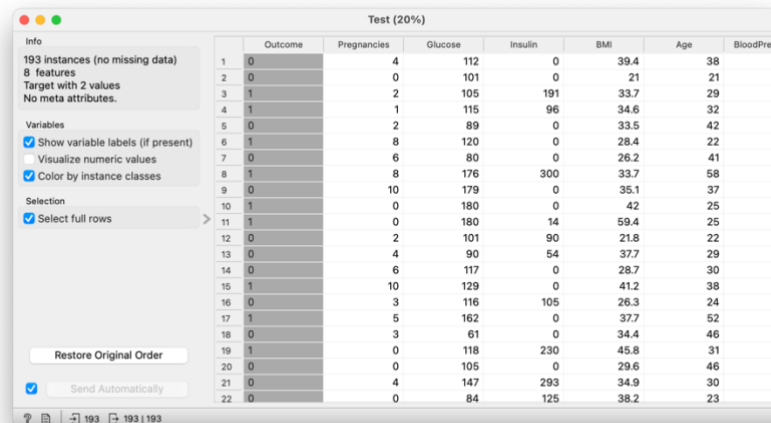
A continuación, podemos acceder al icono **“Train (80%)”** y se abrirá la siguiente ventana en donde podremos observar (en el recuadro rojo de la parte superior izquierda), que tenemos 772 instancias.



	Outcome	Pregnancies	Glucose	Insulin	BMI	Age	BloodPres
1	0	8	107	0	24.6	34	
2	1	12	92	258	27.6	44	
3	1	6	194	0	23.5	59	
4	0	9	123	94	33.1	40	
5	1	4	109	99	34.8	26	
6	1	4	123	0	32	35	
7	0	2	129	0	28	27	
8	1	3	139	0	25.6	22	
9	1	5	144	285	32	58	
10	0	7	83	71	29.3	36	
11	1	0	95	36	37.4	24	
12	1	7	184	0	35.5	41	
13	1	9	165	0	30.4	49	
14	0	6	123	230	33.6	34	
15	1	7	109	0	35.9	43	
16	1	2	134	0	28.9	23	
17	1	10	168	0	38	34	
18	1	3	158	245	31.6	28	
19	1	3	171	135	33.3	24	
20	1	9	140	0	32.7	45	
21	1	3	129	115	26.4	28	
22	1	2	118	0	42.9	21	

Il·lustració 24

De la misma manera, visualizamos el icono “**Test (20%)**” y observaremos la ventana siguiente en donde observaremos que tenemos 193 instancias en nuestro conjunto de datos de pruebas.



	Outcome	Pregnancies	Glucose	Insulin	BMI	Age	BloodPres
1	0	4	112	0	39.4	38	
2	0	0	101	0	21	21	
3	1	2	105	191	33.7	29	
4	1	1	115	96	34.6	32	
5	0	2	89	0	33.5	42	
6	1	8	120	0	28.4	22	
7	0	6	80	0	26.2	41	
8	1	8	176	300	33.7	58	
9	0	10	179	0	35.1	37	
10	1	0	180	0	42	25	
11	1	0	180	14	59.4	25	
12	0	2	101	90	21.8	22	
13	0	4	90	54	37.7	29	
14	0	6	117	0	28.7	30	
15	1	10	129	0	41.2	38	
16	0	3	116	105	26.3	24	
17	1	5	162	0	37.7	52	
18	0	3	61	0	34.4	46	
19	1	0	118	230	45.8	31	
20	0	0	105	0	29.6	46	
21	0	4	147	293	34.9	30	
22	0	0	84	125	38.2	23	

Il·lustració 25

**Pregunta 3.1** Realiza el mismo proceso anterior adjuntando capturas de pantallas (ilustraciones 23, 24 y 25) donde se vea todos los pasos que has seguido para separar los datos en train(80%) y test(20%), y en donde se vea el nombre del proyecto “NombreApellido1” en la captura del modelo (ilustración 23).

## 3.2 Entrenamiento y evaluación del modelo

Una vez tenemos los conjuntos de entrenamiento y de prueba, vamos a implementar el primero de los modelos, la regresión logística. Para esto, conectaremos un objeto “**Logistic Regression**” a nuestro objeto “**Train (80%)**”, como podemos ver en la siguiente ilustración.

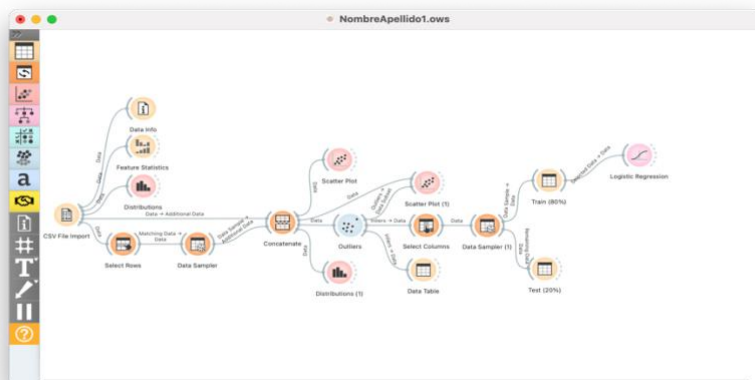


Ilustración 26

Dentro del icono “**Logistic Regression**” podemos configurar las diferentes características de nuestro modelo, aunque para este ejercicio dejaremos los valores por defecto.

A continuación, podemos comprobar cómo ha ido el entrenamiento del modelo de regresión logística, conectando un objeto “**Test and score**” a la salida del objeto “**Logistic regression**” y a la salida del objeto “**Train (80%)**”, como podemos ver en la siguiente ilustración.

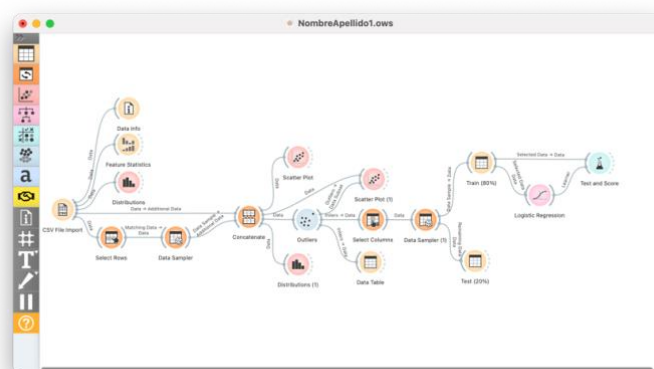


Ilustración 27

Si hacemos clic en el objeto “**Test and score**” se abrirá una ventana como la imagen de la ilustración 28, en donde aparece nuestro modelo y una serie de métricas que permiten

analizar el comportamiento del modelo entrenado al clasificar nuestra clase. En este caso, analizaremos las siguientes métricas:

- **Area Under the Curve (AUC).** Corresponde al área debajo de la curva ROC. Esta curva representa la relación entre la especificidad (precisión) y la sensibilidad (recall). Cuanto más se acerque a 1, mejor.
- **Classification Accuracy (CA).** Nos indica cuántas predicciones hemos hecho correctamente con relación al total de predicciones. Cuanto más se acerque a 1, mejor.
- **F1.** Se trata de una medida que combina precisión (Prec. en la imagen inferior) y recall. Cuanto más se acerque a 1, mejor.

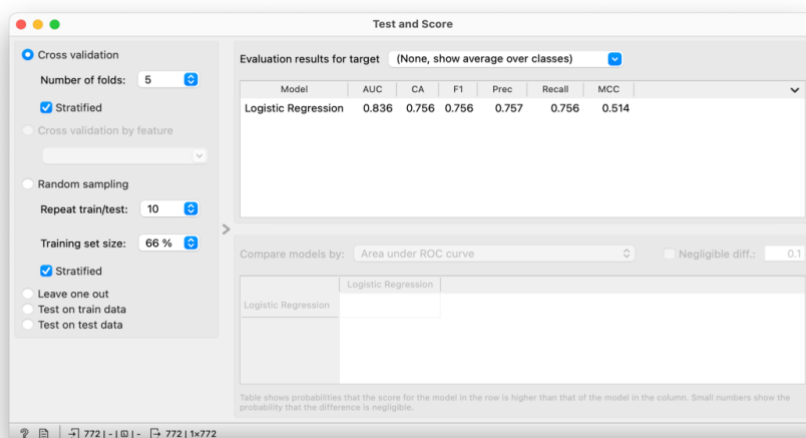
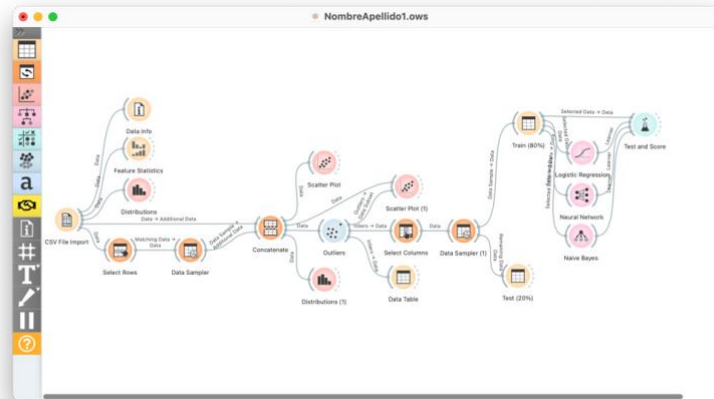


Ilustración 28

Como científicos de datos, a menudo es conveniente probar diferentes modelos antes de escoger uno en concreto, y empezar tareas de optimización sobre este.

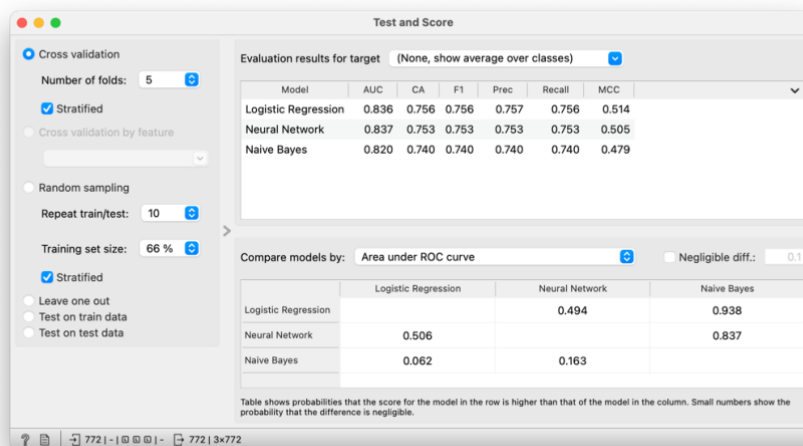
En este sentido, y como ya adelantamos al inicio de la pregunta, vamos a proceder de la misma manera, pero ahora escogeremos dos nuevas técnicas; la **red neuronal** y **Naïve Bayes**.

Procediendo de la misma manera, obtenemos cómo resultado la pantalla de la ilustración 29.



Il·lustració 29

Si ahora hacemos clic en el objeto “**Test and score**” podemos ver las mismas métricas para los tres modelos entrenados.



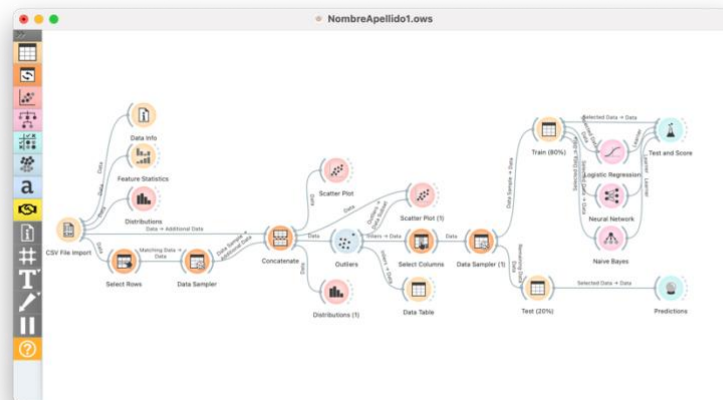
Il·lustració 30

**Pregunta 3.2.A.** Adjunta las capturas pantalla de las de las ilustraciones 29 y 30 y analiza las métricas obtenidas para los tres modelos. ¿Cuál crees que es el mejor modelo?, ¿y el peor? Argumenta tus respuestas. Recuerda que en las capturas donde se vea tu modelo ha de aparecer tu nombre y apellido.

Una vez hecho esto, ya tenemos entrenados nuestros modelos, y el siguiente paso consistirá en ver cuáles de ellos se comportan mejor haciendo predicciones sobre datos anteriormente

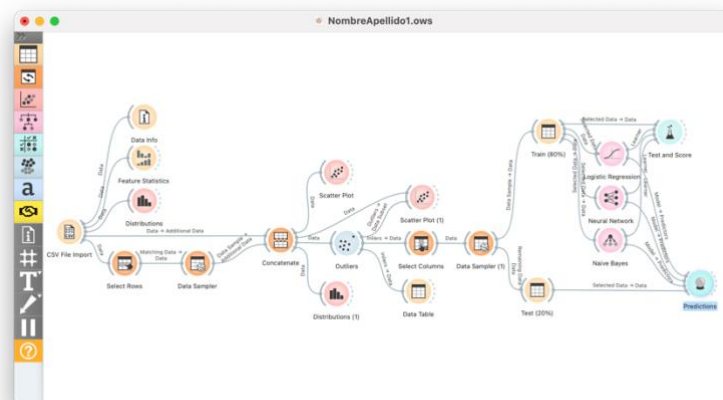
no vistos por los algoritmos: en nuestro caso el subconjunto de datos que hemos reservado para la validación, el conjunto **test (20%)**.

Para esto, conectaremos al conjunto de datos de prueba un objeto “**Predictions**”. El resultado final será cómo el de la siguiente ilustración.



Il·lustració 31

A continuació, y para poder hacer la predicción, conectaremos al nuevo objeto “**Predictions**” los modelos entrenados. El resultado final será como el de la siguiente ilustración.



Il·lustració 32

Si hacemos clic en el objeto “**Predictions**”, nos aparecerá una ventana como la de la ilustración 33, en donde podremos ver las predicciones de las 193 instancias para cada uno

de los modelos (la red neuronal aparece en blanco en la imagen). Por ejemplo, para el modelo “**Logistic Regression**”, la predicción para la instancia ‘1’ es correcta, pues es negativa (‘0’) con un margen de error de casi el 50% respecto a la clase “**outcome**”. La predicción también es acertada para el modelo de **red neural** para esta primera instancia (error del 38%), mientras que el **Naive Bayes** no la predice correctamente (78% de error).

Model	AUC	CA	F1	Prec	Recall	MCC
Logistic Regression	0.838	0.741	0.741	0.741	0.741	0.482
Naive Bayes	0.840	0.751	0.750	0.757	0.751	0.508

Ilustración 33

Para el **análisis comparativo de los modelos**, tenemos las mismas métricas (ver recuadro amarillo de la ilustración 33) que vimos para analizar los modelos de entrenamiento, pero ahora evalúan el comportamiento del modelo con **datos no vistos anteriormente**, simulando el comportamiento en producción del modelo.

Por otro lado, podemos hacer un **análisis de los valores de la clase** (negativo y positivo), de manera individual para cada uno de los modelos, utilizando la **matriz de confusión**. Para esto, conectaremos un objeto “**Confusion matrix**” al objeto “**Predictions**” como en la imagen siguiente.

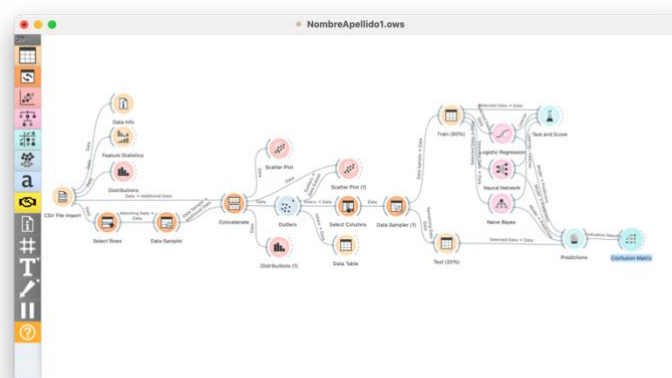


Ilustración 34



Si hacemos clic en el objeto “**Confusion matrix**” se abrirá una ventana como la de la imagen siguiente en donde podemos ver, para cada uno de los modelos (la red neuronal aparece en blanco entre los otros dos modelos), una tabla de dos dimensiones en donde las filas son los valores reales y las columnas los valores predichos.

Un elemento de la matriz representa el número o el porcentaje de instancias (según configuremos la lista desplegable ‘**Show**’ de la esquina superior derecha de la ilustración 35) clasificadas según la clase de la columna, y que realmente pertenecen a la clase de la fila. Sobre esta matriz se calculan las ratios para las métricas anteriormente vistas.

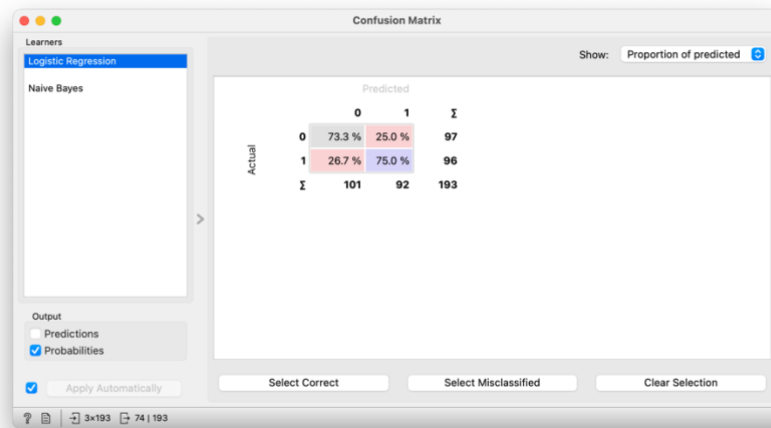


Ilustración 35

En la ilustración anterior, podemos ver cómo la **regresión logística** predice correctamente la clase negativa ('0') con un 73,3% de probabilidad, mientras que la clase positiva ('1') se predice correctamente en un 75%.

**Pregunta 3.2.B.** Repite el proceso realizado anteriormente capturando las pantallas que muestren la predicción de los modelos (pantallas 32 y 33) y realiza un **análisis comparativo de las métricas** obtenidas para cada uno de los modelos que responda a: ¿Qué método tiene mejor comportamiento en base a las métricas analizadas? ¿Cuál tiene el peor comportamiento?

**Pregunta 3.2.C.** Repite el proceso realizado anteriormente para analizar el comportamiento de los modelos para cada uno de los valores ('0' y '1') de la clase '**outcome**', capturando las pantallas que muestren la **construcción del modelo** (ilustración 34) y un **análisis comparativo de las matrices de confusión** de cada uno de los modelos (ilustración 35) que responda a: ¿Cuál es el comportamiento, en la predicción de los valores positivo y negativo de la clase, en cada uno de los modelos?, ¿cuál predice mejor/peor la clase positiva? y ¿cuál predice mejor/peor la clase negativa?

**Importante:** Las capturas que muestren el modelo han de mostrar el "NombreApellido1".



### 3.3 Optimización de modelos

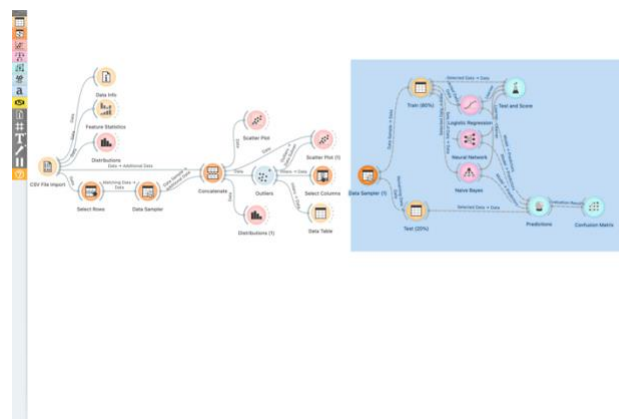
Finalmente, antes de validar completamente un modelo y ponerlo en producción, es importante **optimizar su comportamiento**.

Para conseguir esto, la principal técnica utilizada se conoce con el nombre de **Feature engineering**, la cual permite que los algoritmos de machine learning procesen mejor los datos y establezcan reglas más precisas a partir de las variables o atributos utilizados para la construcción del modelo. Este proceso de optimización, a través de las variables utilizadas, va desde la selección a la transformación de los datos disponibles, por lo que se trata de un aspecto de la ciencia de datos en la que el **factor humano** es clave.

A continuación vamos a realizar este proceso de *featuring engineering* en nuestro modelo.

Cómo hemos visto, la edad es un atributo importante para entender el comportamiento de la enfermedad, pero si lo pensamos detenidamente, ¿estamos realmente interesados en conocer la correlación entre la **edad exacta** y el comportamiento de la enfermedad?, la respuesta es no, no necesitamos la edad exacta, solo saber cómo la edad influye en la enfermedad. Por tanto, no necesitamos el valor numérico o continuo de la variable edad, sino más bien, intervalos o tramos de edad (por ejemplo, jóvenes, adultos, personas mayores...). Este proceso se llama **discretización de variables**, pues estamos transformando un valor continuo o numérico en otro discreto o categórico.

Para hacer esto con *orange3*, vamos a realizar los siguientes pasos. En primer lugar, vamos a eliminar la conexión entre los objetos “**Select columns**” y “**Data sampler**”, a continuación, haremos clic con el ratón en el canvas para seleccionar todos los objetos conectados con el “Data sampler” para poder desplazarlos a la derecha para hacer espacio entre ambos objetos (esto es necesario solamente para hacer espacio entre los objetos “Select columns” y “Data sampler”, en caso contrario no es necesario hacerlo).



*Ilustración 36*

A continuación, conectamos al objeto “**Select Columns**” un nuevo objeto llamado “**Discretize**”, y a su vez este último lo conectamos al objeto “**Data Sampler**”, siendo el resultado final como el de la imagen siguiente.

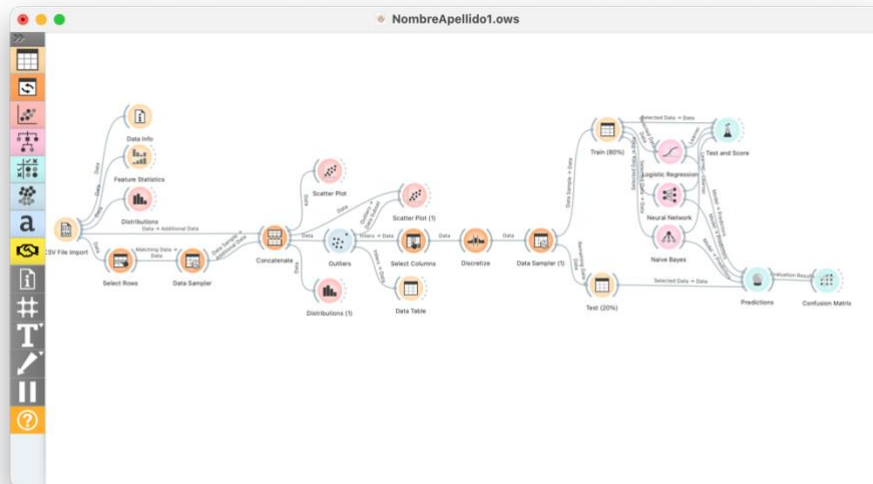


Ilustración 37

Acto seguido, hacemos clic en el objeto “**Discretize**” y se abrirá una pantalla como la de la ilustración 38, en donde seleccionaremos el atributo **Age** y el método de discretización “**Fixed width**” con el valor de 10, lo que nos categoriza el atributo en tramos de 10 años.

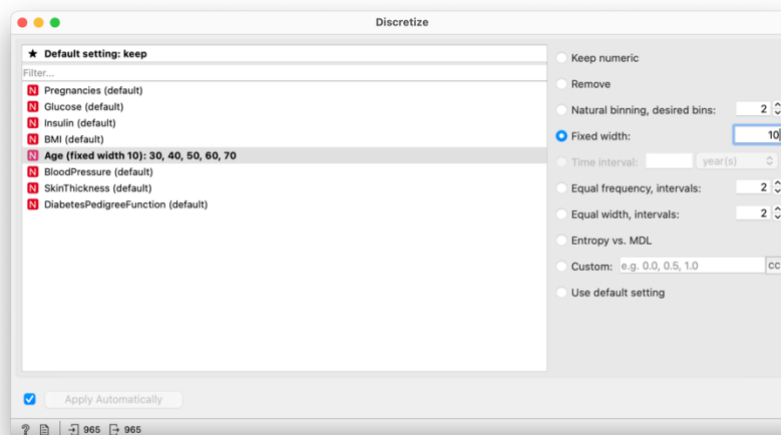
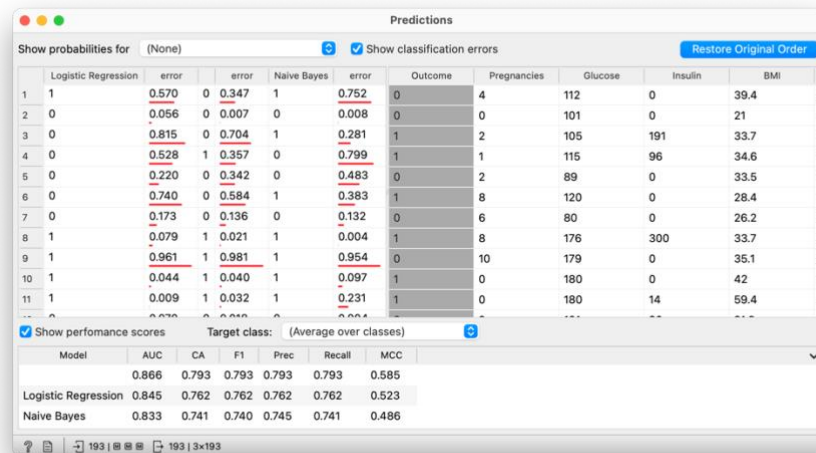


ilustración 38

Finalmente, si hacemos clic en el objeto “**Predictions**” podemos analizar el comportamiento del modelo una vez realizado este cambio.



	Logistic Regression	error	error	Naive Bayes	error	Outcome	Pregnancies	Glucose	Insulin	BMI
1	1	0.570	0	0.347	1	0.752	0	112	0	39.4
2	0	0.056	0	0.007	0	0.008	0	101	0	21
3	0	0.815	0	0.704	1	0.281	1	105	191	33.7
4	0	0.528	1	0.357	0	0.799	1	115	96	34.6
5	0	0.220	0	0.342	0	0.483	0	89	0	33.5
6	0	0.740	0	0.584	1	0.383	1	120	0	28.4
7	0	0.173	0	0.136	0	0.132	0	80	0	26.2
8	1	0.079	1	0.021	1	0.004	1	176	300	33.7
9	1	0.961	1	0.981	1	0.954	0	179	0	35.1
10	1	0.044	1	0.040	1	0.097	1	180	0	42
11	1	0.009	1	0.032	1	0.231	1	180	14	59.4

Model	AUC	CA	F1	Prec	Recall	MCC
Logistic Regression	0.866	0.793	0.793	0.793	0.793	0.585
Naive Bayes	0.833	0.741	0.740	0.745	0.741	0.486

Il·lustració 39

Si comparamos el resultado obtenido (ilustración 39) con el anterior (ilustración 33), y la matriz de confusión nueva con la anterior (ilustración 35) observamos como las métricas han mejorado, por lo que deducimos que el cambio realizado ha eliminado parte del “ruido” introducido por el valor numérico de la variable edad.

**Pregunta 3.3.A** Repite el proceso anterior, adjuntando las capturas de las pantallas 37, 38 y 39, y analiza los cambios en las nuevas métricas obtenidas y en la matriz de confusión.

**Pregunta 3.3.B** Prueba de optimizar el modelo actual jugando con el objeto **Discretize**. Para esto, usa la variable y un método de tu elección. Haz captura de las pantallas 38 y 39, argumentando los resultados obtenidos.

**Importante:** La captura de la ilustración 37 ha de mostrar tu NombreApellido1.

## Criterios de evaluación

- La pregunta 3.1 se valorará con **0.5 puntos** como máximo.
- Los apartados 3.2.A, 3.2.B y 3.2.C se valorarán con **0.5 puntos** como máximo cada una, con un total de **1.5 puntos** como máximo para la pregunta 3.2.
- La pregunta 3.3 se valorará con **1 punto** como máximo. Los apartados 3.3.A y 3.3.B se valorarán con **0.5 puntos** como máximo respectivamente.
- Todas las capturas que incluyan el modelo implementado deben mostrar el **“NombreApellido1” claramente visible**. Las que no lo tengan no serán valoradas.
- Todas las capturas de pantalla se tendrán que **argumentar debidamente**, tanto el contenido como el proceso realizado para su obtención. En caso contrario, se penalizará reduciendo la nota máxima a la mitad.
- El **tamaño de las capturas** de pantalla ha de facilitar su análisis, y en caso de no poderse analizar, no se valorarán.
- Extensión **mínima** por pregunta: **100 palabras**. Las respuestas que no cumplan con los requisitos mínimos de número de palabras se penalizarán, reduciendo la nota máxima a la mitad.
- Las capturas que muestren el modelo y **no incluyan el “NombreApellido1”** no se valorarán.

## Pregunta 4: Machine Learning III (20%)

### Enunciado

Como hemos visto en los apartados anteriores, una de las métricas más relevantes para considerar la bondad de un modelo es la **métrica AUC-ROC** o **área bajo la curva ROC**. Se trata de una medida relacionada con el equilibrio del dataset. Esta métrica es una representación gráfica de la relación entre la **especificidad** y la **sensibilidad** de un modelo.

**Pregunta 1.** Investiga en internet para definir y explicar con tus propias palabras y fórmulas los conceptos que sirven para definir la AUC-ROC:

- 1.1. La **sensibilidad (recall en inglés)** o **razón de verdaderos positivos (VPR)**.
- 1.2. La **especificidad** o **razón de verdaderos negativos (VNR)**.
- 1.3. ¿A qué responden cada uno de estos conceptos en el caso de uso que hemos visto para la detección de una enfermedad?

La otra métrica que hemos visto, relacionada con el equilibrio del dataset, es la **F1**, muy usada para analizar la relación “**precision**” y “**recall**” (o sensibilidad)

**Pregunta 2.** Investiga en internet para definir y explicar con tus propias palabras y fórmulas, en el contexto del caso detección de la diabetes:

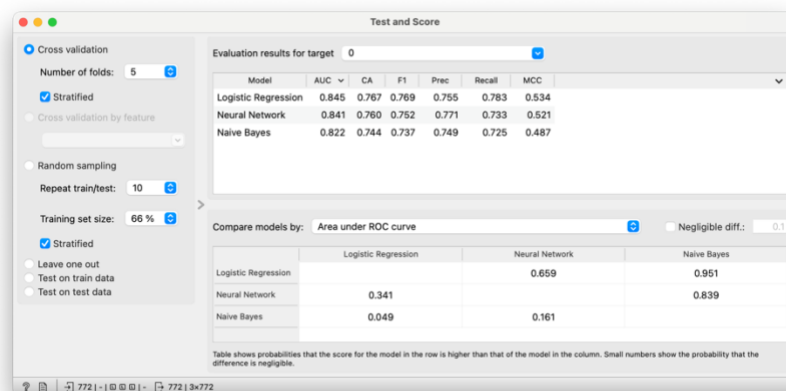
- 2.1. La **precisión**.
- 2.2. ¿Cómo combina la métrica F1 los dos conceptos vistos anteriormente de precisión y sensibilidad (recall)?

En función de los valores de sensibilidad y precisión, obtenidos en *Orange3* para cada uno de los valores de la clase ('0' y '1') de nuestro modelo de entrenamiento, y sabiendo que, en función de la combinación de valores de las métricas tenemos los siguientes escenarios:

- Alta precisión y alto recall el modelo maneja perfectamente esa clase.
- Alta precisión y bajo recall el modelo no detecta la clase muy bien, pero cuando lo hace es altamente fiable.
- Baja precisión y alto recall: el modelo detecta bien la clase, pero también incluye muestras de otras clases.
- Baja precisión y bajo recall el modelo no logra clasificar la clase correctamente.

Supón que consideramos un valor alto a partir del 75% (0.75).

Si a continuación hacemos clic en el objeto **Test and Score**, correspondiente al entrenamiento de los modelos y filtramos la lista desplegable **Evaluation results for target**, para el valor negativo de la clase (0) visualizamos una pantalla como la de la siguiente ilustración.



Il·lustració 40

**Pregunta 3.** Adjunta dos capturas de pantalla, una para cada uno de los valores de la clase (ilustración 40). ¿Cuál es el comportamiento de los modelos entrenados, en función de ambas métricas, para cada uno de los valores de la clase ('0','1') del dataset?

**Nota:** Para la respuesta, utiliza una tabla para cada valor de la clase como la siguiente para responder.

MODELO	Clase Negativa (0)		
	Precision	Recall	Análisis
Logistic Regression			
Neural Network			
Naive Bayes			

## Criterios de evaluación

- La pregunta 1 se valorará con **0.75 puntos** como máximo, y cada uno de los tres apartados con **0.25 puntos** respectivamente como máximo.
- La pregunta 2 se valorará con **0.50 puntos** como máximo, y cada uno de los dos apartados con **0.25 puntos** como máximo.
- La pregunta 3 se valorará con **0.75 puntos** como máximo, **0.375 puntos** por cada una de las dos tablas como máximo.
- Las respuestas tienen que estar argumentadas y mostrar opinión crítica analítica, en caso contrario se penalizará, reduciendo la nota máxima a la mitad.