TUTORIAL PARA LA CONSTRUCCIÓN DE MODELOS QUIMIOMÉTRICOS Y DE APRENDIZAJE AUTOMÁTICO CON ORANGE, UN SOFTWARE LIBRE Y GRATUITO

Ángel Sánchez Illana, David Pérez Guaita

Departamento de Química Analítica, Facultad de Química, Universidad de Valencia, C/ Dr. Moliner 50, 46100 Burjassot, Valencia

1. Introducción

En el campo de la química analítica el análisis de datos juega un papel fundamental. En este sentido, no somos ajenos a los rápidos avances en computación que revolucionan nuestra capacidad de procesar y analizar gran cantidad de información. Esto explica el surgimiento de la quimiometría hace más de medio siglo siendo hoy en día un campo bien establecido dentro de la química analítica [1].

En este contexto, en los últimos años, el surgimiento y auge de la inteligencia artificial, y el aprendizaje automático (*machine learning*) está teniendo un gran impacto. Una búsqueda bibliográfica (noviembre 2023) en todas las bases de datos de la *Web of Science* con combinaciones de las palabras clave "*analytical chemistry*", "*chemometrics*", *"machine learning", "deep learning" y "artificial intelligence"* pone de manifiesto la creciente popularidad del *machine learning* (Figura 1). Mientras que las publicaciones en quimiometría se han mantenido más o menos estables tras un *boom* en los años 1990, las nuevas palabras clave han ganado popularidad en el último decenio, creciendo de manera aparentemente independiente. En efecto, el desarrollo de plataformas analíticas que proporcionan multitud de variables químicas, además de la miniaturización de computadoras capaces de realizar complejos algoritmos, está permitiendo que la química analítica aproveche las tendencias del *big data* y la inteligencia artificial.

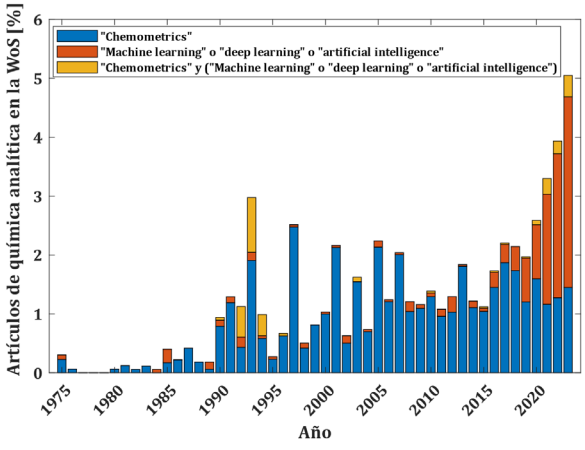

Figura 1. Porcentaje de artículos con la palabra clave "*analytical chemistry*" que comparten las palabras clave "*chemometrics*" y/o "*machine learning*" y relacionadas desde el año 1975. Fuente: Web of Science.

Más allá de las tendencias actuales, existe cierto debate en la comunidad acerca de la distinción entre quimiometría y *machine learning*. Algunos autores consideran que la primera se enfoca principalmente en relaciones lineales entre las variables mientras que la segunda trabaja con grandes sets de datos no lineales sin suponer una estructura de los datos y necesitando un proceso de entrenamiento previo para modelizarla. Así, métodos de proyección basados en el uso de combinaciones lineales como el análisis de componentes principales (PCA) o el análisis de mínimos cuadrados parciales (PLS) son considerados métodos quimimétricos, mientras que métodos como los árboles de decisión, los *random forest* (RF), las redes neuronales artificiales (ANN) o las máquinas de vectores de soporte (SVM) son considerados métodos de aprendizaje automático.

En cualquier caso, un desafío para la enseñanza y el uso de la quimiometría y el aprendizaje automático es la falta de conocimientos avanzados en programación y estadística, lo que resulta en una comprensión superficial y un posterior rechazo. Sin embargo y paradójicamente, el estudiantado de química analítica muestra un gran interés en aprender *machine learning* y quimiometría y parcialmente reconocen su utilidad. Hemos observado estas tendencias en nuestros estudiantes de máster [2], como se ilustra en la Figura 2.

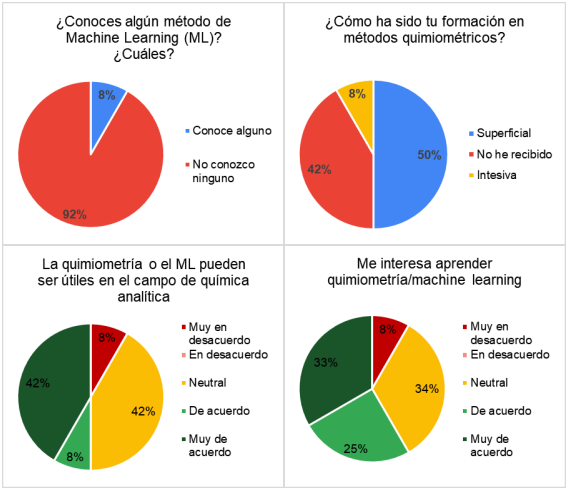

Figura 2. Extracto de algunos resultados de nuestras encuestas realizadas en un máster de química.

A lo largo del tiempo, se han hecho esfuerzos para cerrar esta brecha entre programación y química analítica mediante el empleo de software interactivo y visual para implementar modelos quimiométricos y de aprendizaje automático. Actualmente contamos con numerosas herramientas de este tipo, pero muchas de ellas son de pago como Unscrambler de Camo Analytics (ahora AspenTech) o Solo y PLS_Toolbox de Eigenvector Research. Otras opciones son programas escritos en MATLAB (también de pago), R o Python que requieren algún conocimiento, aunque mínimo, de programación [2–4].

En este contexto, investigadores de la Universidad de Ljubljana (Eslovenia) han desarrollado Orange, un software interactivo, gratuito y de código abierto [5]. Orange utiliza la programación visual, es decir, la organización de componentes en flujos de trabajo (*workflows*) mediante una interfaz gráfica como se muestra en la Figura 3. Cada componente, llamado *widget*, incorpora alguna tarea como importar datos, preprocesamiento, visualización, modelización, evaluación o predicción. Orange permite a los usuarios construir estos flujos de trabajo de manera gráfica, conectando los *widgets* de acuerdo con sus necesidades. Orange dispone de una amplia biblioteca de *widgets* incluyendo algunos adicionales orientados a temas específicos disponibles a través de complementos (*add ons*). Asimismo, Orange dispone de una extensa documentación, una comunidad activa de usuarios organizada en canales de Discord y StackExange además de un canal de Youtube con multitud de contenido [6,7].

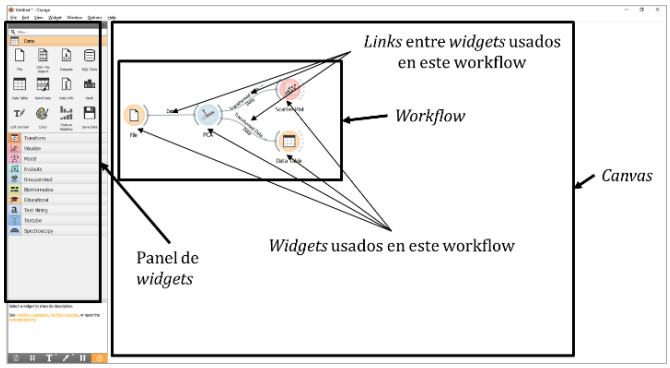

Figura 3. Ventana de Orange con un *workflow* para construir un PCA en el *canvas*. Indicación de los principales elementos.

El objetivo de este artículo es introducir un tutorial de Orange que explore la capacidad de Orange para trabajar con datos químicos con fines docentes y de investigación. Presentaremos algunas funcionalidades e usando diferentes ejemplos de métodos quimiométricos y de *machine learning*, así como ejercicios prácticos comúnmente utilizados en cursos introductorios de quimiometría. Todo el contenido de este tutorial y los ejercicios resueltos se encuentran disponibles en el repositorio Zenodo bajo licencia Creative Commons CC BY-NC-SA 4.0 [8].

No entraremos en aspectos teóricos en este tutorial por lo que recomendamos los trabajos de Brereton [1,9–11] y la extensa compilación editada por Brown, Tauler y Walczak [12] enfocados en quimiometría y reconocimiento de patrones aplicados a datos químicos. Para un curso multidisciplinar introductorio orientado en el *machine learning* recomendamos el trabajo de Abu-Mostafa et al. [13].

2. Importando datos y creando un *workflow*

Tras la instalación de Orange, una vez descargado desde su sitio web oficial (https://orangedatamining.com/), lo primero que haremos es instalar los *Add-ons* de "*Bioinformatics", "Educational"* y *"Spectroscopy"*. Esto puede hacerse desde Options>Add ons. Se pueden instalar en cualquier momento tantos *Add ons* como se necesiten. A continuación, ya podemos construir nuestros primeros *workflows*.

Todo *workflow* siempre empieza con un *widget* de importación de los datos. Podemos importar datos a Orange directamente utilizando el *widget* "*File*" desde multitud de fuentes como hojas de cálculo (p.ej. .xls, .xlsx…), ficheros de texto plano (p.ej. .txt, .csv, .tsv…), o ficheros más complejos como variables de MATLAB (.mat) o datos espectrales de diferentes casas comerciales (p.ej. Bruker o Agilent). También tenemos el *widget* "*CSV File Import*" que nos permite importar desde ficheros de texto plano .csv con opciones avanzadas como ajustar el carácter delimitador o la codificación de los caracteres. Estos widgets de importación nos permiten también seleccionar el tipo de variable (*numeric*, *categorical…*) y el rol que va a tener en nuestro *workflow* (*feature, target, metadata…).* Existen otros *widgets* para importar datos más sofisticados y también el *widget* "*Dataset"* que nos permite acceder a una colección de sets de datos con los que podemos practicar. Asimismo, los *widgets* "*Select Columns"* y *"Edit Domain"* nos permiten modificar en cualquier momento el tipo de cada variable y su rol.

Para colocar *widgets* en el *canvas* únicamente tenemos que seleccionar el que queramos del panel de la izquierda y arrastrarlo al *canvas*. También podemos seleccionar widgets haciendo clic derecho en mitad del *canvas* y escribiendo en el menú que aparece el nombre del *widget* que queremos. Cada *widget* puede unirse con otro por sus partes de la izquierda o derecha y moverse por el *canvas* solo o en grupo, unido o suelto. Por la parte de la izquierda se le "alimenta" con datos (*inputs*) a los que aplica su tarea y por la derecha devuelve el resultado (*outputs*) hacia otros *widgets*. Así, el flujo va de izquierda a derecha. No todos los *widgets* son compatibles entre sí, como es lógico.

Al hacer doble clic en el *widget* se nos muestra su salida y entrada y se nos permite configurar cómo la tarea es aplicada y seleccionar cómo y qué datos queremos que sean devueltos. Algunos *widgets* aplican un preprocesado a los datos por defecto, este puede comprobarse en la documentación de cada *widget* y editarse posteriormente como veremos después. Hay algunos *widgets* que tienen más de una entrada y salida. Si hacemos doble clic en los

enlaces entre *widgets* podemos seleccionar qué entrada y salida queremos utilizar. También podemos cambiar la etiqueta de cada *widget* para identificarlos mejor con la opción "*Rename*" al hacer clic derecho en ellos. Veremos diferentes maneras de conectar los *widgets* en los diferentes ejemplos a lo largo del tutorial.

Orange, por defecto, ejecuta el *workflow* a tiempo real conforme vamos haciendo las uniones, pero podemos pausar esa ejecución. Esto puede ser muy útil cuando trabajamos con cálculos complejos o *workflows* muy grandes para evitar pérdidas de rendimiento de la aplicación. Esto podemos hacerlo usando Shift+F o haciendo clic en el símbolo de pausa de la esquina inferior izquierda. En la Figura 4 se muestran diferentes ejemplos de *workflows* en los que se importan datos y se visualizan con el *widget "Data Table"*.

Figura 4. a) *Workflow* para importar datos, b) tabla con los datos correctamente importados.

Ejercicio 2.1. Reproduce los *workflows* de la Figura 4 para importar los datasets "fastfood.csv" y "water_potability.xlsx" y renombra los diferentes *widgets* y enlaces.

3. Visualización de datos y análisis exploratorio

Orange ofrece diferentes *widgets* muy potentes para realizar visualización de datos. Además, nos permite ir actualizando estas visualizaciones conforme vamos modificando los *widgets* que se encuentran conectados "aguas arriba". De esta manera si colocamos un *widget* de visualización después de un *widget* de tabla en el que hemos seleccionado unas columnas o filas podemos mandar a la visualización esos datos. De igual manera si seleccionamos en una visualización un conjunto de datos podemos mandar esa selección a otro *widget*.

En la Figura 5 se muestra una pantalla con un workflow para la construcción de un *scatter plot*. Como se aprecia en la figura, Orange permite controlar el tamaño de los elementos y la manera de colorearlos entre otros parámetros. Las figuras obtenidas pueden exportarse fácilmente a diferentes formatos como .jpg o .svg. a partir de la opción "*Save Image*" de la esquina inferior izquierda.

También tenemos muchos otros *widgets* de visualización entre los que destacamos "*Distributions*", "*Box plot*", "*Violin plot*" y "*Sieve Diagram*". Con estos *widgets* podemos calcular la media, la mediana, los intervalos de confianza, realizar algunas pruebas estadísticas y representar la distribución de nuestras variables como mostramos en la Figura 6.

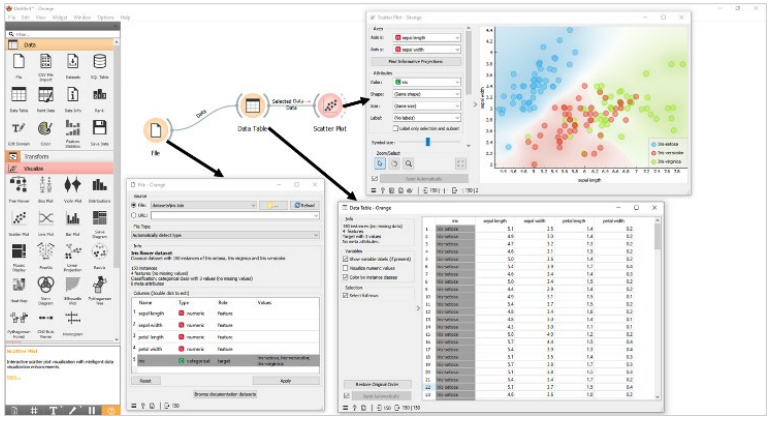

Figura 5. Ventana de Orange con un *workflow* para construir *scatter plot* con el set de datos Iris de R. Fisher.

Figura 6. Ventana de Orange con un *workflow* para construir diferentes visualizaciones de datos a partir del set de datos *Wine*.

Ejercicio 3.1. Construye un *workflow* para importar el dataset Iris de R. Fisher y representar un scatter plot con un par de variables.

Ejercicio 3.2 Crea un *workflow* para reproducir las visualizaciones de la Figura 6 con el set de datos *Wine* importándolo con el *widget* "*Dataset*". Personaliza las figuras y expórtalas en formato .svg.

4. Agrupamiento jerárquico

El primer método de análisis quimiométrico que construiremos con Orange es el agrupamiento jerárquico o *hierarchical clustering* utilizando los *widgets* "*Distances*" y "*Hierarchical Clustering*". Con el primero calculamos las distancias entre columnas (o filas) de nuestra matriz de datos utilizando diferentes métricas y normalizaciones. La salida de este widget contiene la matriz de distancias que puede utilizarse para construir un dendrograma con el *widget* "*Hierarchical Clustering*". Un *workflow* de este tipo se muestra en la Figura 6 para el set de datos *Wine*.

Ejercicio 4.1 Reproduce el *workflow* de la Figura 6 incorporando algún *widget* de visualización a la salida del *widget* "*Hierarchical Clustering*" para representar las distribuciones de cada clúster.

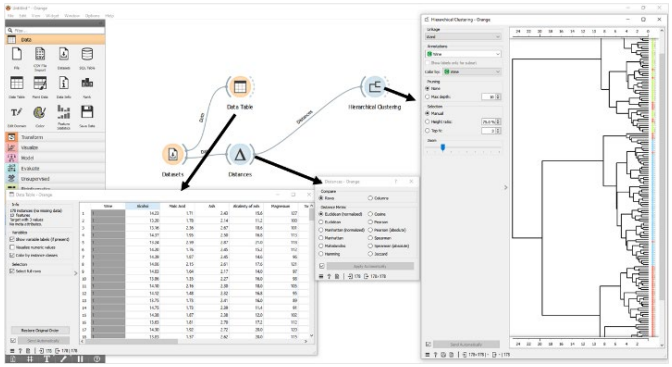

Figura 6. Ventana de Orange con un *workflow* para construir un análisis de agrupamiento jerárquico empleando el set de datos *Wine*.

5. Preprocesado y análisis de componentes principales de datos espectrales

El análisis de componentes principales (PCA) es una de las herramientas más utilizadas en quimiometría para el análisis exploratorio no supervisado. Con Orange, podemos construir PCA con datos químicos como tablas de señales analíticas utilizando el widget "PCA". Una vez añadido al *workflow,* este nos permite seleccionar el número de componentes principales inspeccionando a la vez el % de varianza explicada.

En cuanto a las matrices espectrales, pueden trabajarse con Orange con los *widgets* disponibles en el *Add on* "*Spectroscopy".* El widget "*Spectra*" nos permite representar como espectros las matrices de datos importadas y colorearlos por categorías como se muestra en la Figura 7.

Usando "*Preprocess Spectra*", podemos aplicar diversas herramientas de preprocesamiento, detalladas en la Figura 8 a). Cada opción en el menú permite la configuración específica de parámetros, incluyendo recorte de regiones espectrales, normalización y realización de derivadas.

También tenemos numerosas opciones de preprocesado para trabajar con matrices de datos no espectrales disponibles en el *widget* "*Preprocess"* como se muestran en la Figura 8 b). En este *widget* tenemos preprocesados que nos permiten aplicar escalados, normalizaciones, eliminar o imputar datos, entre otros. Si unimos *Preprocess* a otro *widget* como "*preprocessor*" nos permite modificar su preprocesado por defecto.

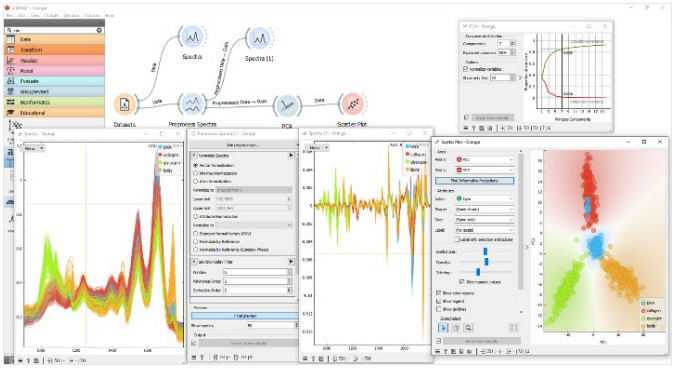

Figura 7. Ventana de Orange con un *workflow* para preprocesar y construir un PCA con datos espectrales del set de datos *Liver spectroscopy (Collagen)*.

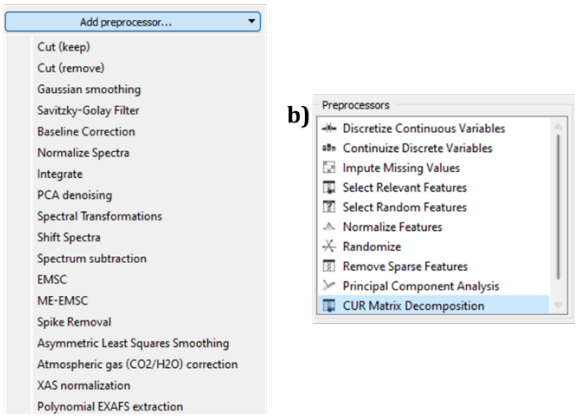

Figura 8. Lista de preprocesamientos disponibles, a) en el *widget* "*Preprocess Spectra"* y b) en el *widget* "*Preprocess*".

Ejercicio 5.1. Construye un *workflow* para realizar un PCA con datos espectrales como el de la Figura 7. Importa el set de datos de Orange "*Liver spectroscopy (Collagen)*" y aplica los preprocesados espectrales de normalización y derivatización que consideres oportunos. Extrae los diagramas de puntuaciones (*scores)* y cargas (*loadings*).

Ejercicio 5.2. Construye un *workflow* para realizar un PCA con datos químicos de los sets de datos "*Wine*" de Orange y "water_potability.xlsx". Aplica los preprocesados espectrales de normalización y derivatización que consideres oportunos.

6. Análisis de mínimos cuadrados parciales y validación de modelos

En el ámbito de la quimiometría, la regresión de mínimos cuadrados parciales (PLS) destaca como método de predicción de concentraciones u otras propiedades de las muestras (variables predictoras) a partir de señales químicas (variables respuesta).

A la hora de construir cualquier modelo supervisado, resulta fundamental realizar correctamente la selección de los parámetros (p. ej. el preprocesado y el número de variables latentes en un PLS) óptimos del modelo evitando el sobreajuste. En este sentido, todo modelo ha de validarse con un set de datos no utilizado en su construcción (calibración). Tanto para la selección de parámetros como para la validación del modelo, podemos realizar diferentes tipos de validación cruzada mediante el *widget "Test and Score"*. Así, podemos calcular los errores de predicción sobre el set de validación cruzada para diferentes métodos de predicción o combinaciones de parámetros como se muestra en la Figura 9.
Los modelos, una vez construidos (es decir, calibrados) con aplicación automática de la validación cruzada, pueden aplicarse a sets de datos externos utilizando el *widget* "*Predictions*" como se muestra en la Figura 10. En este *workflow*, como se puede observar, también se obtiene el gráfico compara los valores de variable predicha vs. variable medida y se obtiene el vector de regresión en forma de espectro, tras aplicarle el widget "*Transpose*".

Hay que tener en cuenta que el preprocesado aplicado a los datos utilizados para construir el modelo queda guardado en el modelo y se aplica a los datos crudos que son utilizados en la predicción. Es por ello por lo que no se coloca el *widget* de preprocesado en los datos crudos usados para predecir.

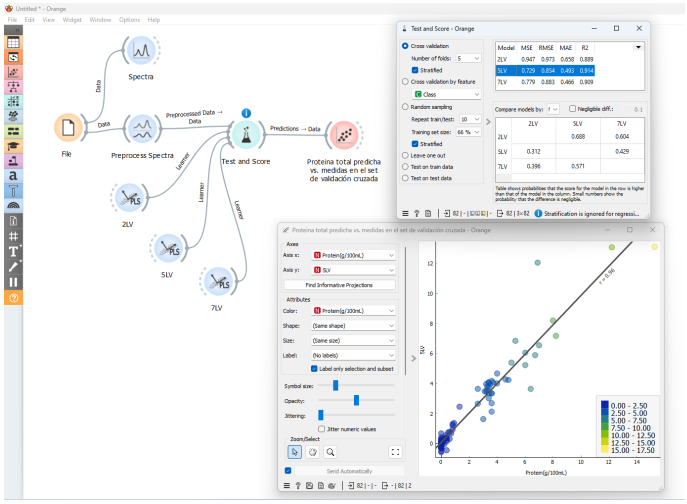

Figura 9. Uso del *widget* "*Test and Score*" para realizar una validación cruzada con datos espectrales y comparar diferentes combinaciones de parámetros.

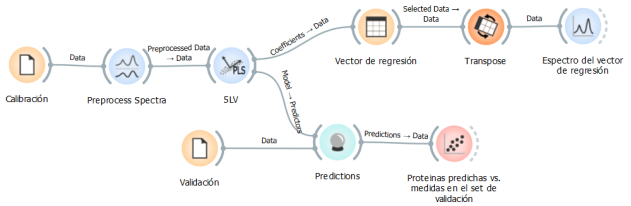

Figura 10. Ejemplo de *workflow* para construir y aplicar un PLS a datos espectrales para la predicción de la concentración de proteínas totales en muestras de alimentos.

Ejercicio 6.1. Construye un modelo PLS para predecir las proteínas totales a partir de datos espectrales. Genera la regresión proteína predicha vs. proteína medida y el vector de regresión. Aplica diferentes estrategias de validación y predicción en datos con diferentes preprocesados: a) Optimiza los parámetros del modelo mediante validación cruzada en el conjunto de datos "Proteins_Cal.csv"; b) Valida el modelo óptimo con el conjunto de datos "Proteins_Val.csv" y calcula el error de predicción; c) Emplea el widget "*Test and Score*" para construir un modelo con el 60% de las muestras y valida con el 40% restante utilizando el conjunto de datos "Proteins_Cal_Val.csv".

7. Análisis mediante algoritmos de *machine learning*

Como ejemplos de métodos de *machine learning* aplicados a la química analítica que pueden implementarse con Orange, en este tutorial trabajaremos con árboles de decisión, *random forests,* y *support vector machines*.

En la Figura 11 se muestra un *workflow* para construir un árbol de decisión con el set de datos "*Wine*". Como puede apreciarse en la figura, puede combinarse con un *scatter plot* de manera similar al análisis mediante agrupamiento jerárquico. Los parámetros del modelo pueden optimizarse utilizando validación cruzada como hemos visto anteriormente y también podemos utilizar el *widget "Confusion Matrix"* para obtener los errores de clasificación correspondientes y "*ROC Analysis*" para calcular la curva ROC de cada predicción.

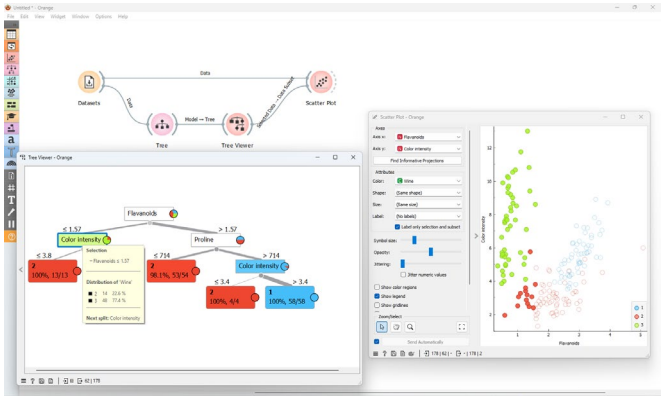

Figura 11. Ejemplo de *workflow* para construir un árbol de decisión con el set de datos "*Wine*".

El método de *random forests* puede considerarse una generalización de los árboles de decisión, ya que construye un "*ensemble*" o conjunto de múltiples árboles de decisión durante el proceso de entrenamiento. El *widget* "*Radom Forest*" nos permite modelizarlos con Orange de manera similar a como se muestra en la Figura 11 y combinarse con el resto de *los widgets* de validación y predicción que hemos visto. Como peculiaridad de los árboles de decisión y los *random forest,* tenemos los *widgets* "*Pytagorean Tree*" y "*Pytagorean Forest*", respectivamente, que nos permiten visualizar los modelos de una manera más compacta que

con árboles y son muy útiles para estudiar la estructura del modelo.

El último método que vamos a trabajar es el SVM. Orange implementa la libraría LIBSVM en el *widget* "SVM". A través de este *widget* tenemos acceso a los hiperparámetros del SVM, que podemos seleccionar y optimizar manualmente. Lamentablemente, no podemos automatizar esta optimización, pero sí podemos construir combinaciones de validación y predicción al igual que hemos hecho hasta ahora y seleccionar las mejores combinaciones manualmente.

Ejercicio 7.1 Construye un *workflow* para crear un modelo de árbol de decisión como el de la Figura 11 y un *random forest* con el set de datos "glass.csv". Compara mediante los *widgets* "*Test and Score*", "*ROC Analysis*" y "*Confusion Matrix*" el desempeño de las diferentes combinaciones de parámetros de ambos modelos.

Ejercicio 7.2. Incluye SVM entre los modelos a utilizar en el ejercicio anterior y optimiza los parámetros del SVM hasta obtener la mejor clasificación.

8. Reflexiones finales

Orange es un gran recurso para la quimiometría y el *machine lerarning* en química analítica. Como hemos visto, podemos construir modelos complejos sin necesidad de saber programación. Asimismo, su potencial para usarse en clase como material de apoyo es enorme y ya ha demostrado ser útil en asignaturas de química analítica en nuestra Universidad [2].

Este tutorial se ha limitado a unos pocos ejemplos clásicos y sólo hemos utilizado una pequeña porción de los *widgets* que ofrece Orange. La versión actual (3.36.1) contiene muchos más y, debido a su naturaleza de código abierto, seguramente vengan más en el futuro en función de las necesidades de la comunidad. Además, existe la posibilidad de incluir scripts de Python en los propios *workflows* por lo que las posibilidades son virtualmente ilimitadas para adaptar estos *workflows* a nuestros problemas analíticos. Animamos a la comunidad de química analítica a probar Orange, explorar su detallada documentación y consultar su canal de Youtube [6].

9. Agradecimientos

Los autores agradecen al Vicerrectorado de Formación Permanente, Transformación Docente y Empleo de la Universitat de València por la financiación del proyecto de innovación educativa emergente LEARNAQA [cod. 2735964]. ASI agradece al Ministerio de Universidades del Gob. de España por la ayuda Margarita Salas [cod. UP2021-044-MS21-084], financiada por la Unión Europea, NextGenerationEU. DPG agradece al MCIN/AEI/10.13039/501100011033 por su ayuda Ramón y Cajal [cod. RYC2019-026556-I].

10. Bibliografía

[1] R.G. Brereton, J. Jansen, J. Lopes, F. Marini, A. Pomerantsev, O. Rodionova, J.M. Roger, B. Walczak, R. Tauler, Chemometrics in analytical chemistry—part I: history, experimental design and data analysis tools, Anal. Bioanal. Chem. 409 (2017) 5891–5899. https://doi.org/10.1007/s00216-017-0517-1.

[2] Á. Sánchez-Illana, B. Wood, D. Pérez-Guaita, Enseñanza del machine learning y la quimiometría en química analítica mediante propuestas prácticas e interactivas, en: In-RED 2023 IX Congr. Innov. Educ. Docencia En Red, 2023. https://doi.org/10.4995/INRED2023.2023.16679.

[3] D. Perez-Guaita, Z. Richardson, A. Rajendra, H.J. Byrne, B. Wood, From bench to worktop: Rapid evaluation of nutritional parameters in liquid foodstuffs by IR spectroscopy, Food Chem. 365 (2021) 130442. https://doi.org/10.1016/j.foodchem.2021.130442.

[4] T.M. Antonelli, A.C. Olivieri, Developing and Implementing an R Shiny Application to Introduce Multivariate Calibration to Advanced Undergraduate Students, J. Chem. Educ. 97 (2020) 1176–1180. https://doi.org/10.1021/acs.jchemed.9b00850.

[5] J. Demsar, T. Curk, A. Erjavec, C. Gorup, T. Hocevar, M. Milutinovic, M. Mozina, M. Polajnar, M. Toplak, A. Staric, M. Stajdohar, L. Umek, L. Zagar, J. Zbontar, M. Zitnik, B. Zupan, Orange: Data Mining Toolbox in Python, J. Mach. Learn. Res. 14 (2013) 2349–2353. https://jmlr.org/papers/v14/demsar13a.html

[6] Orange Data Mining - Canal de YouTube. https://www.youtube.com/channel/UClKKWBe2SCAEyv7ZNGhIe4g .

[7] B.L. Ljubljana University of, Orange Data Mining - Documentation, Orange Data Min. https://orangedatamining.com.

[8] Á. Sánchez Illana, D. Pérez-Guaita, Tutorial para la construcción de modelos quimiométricos y de aprendizaje automático con Orange, un software libre y gratuito: Ejercicios resueltos. https://doi.org/10.5281/zenodo.10155451.

[9] R.G. Brereton, J. Jansen, J. Lopes, F. Marini, A. Pomerantsev, O. Rodionova, J.M. Roger, B. Walczak, R. Tauler, Chemometrics in analytical chemistry—part II: modeling, validation, and applications, Anal. Bioanal. Chem. 410 (2018) 6691–6704. https://doi.org/10.1007/s00216-018-1283-4.

[10] R.G. Brereton, Chemometrics: Data Driven Extraction for Science, 2nd ed., Wiley, 2018. https://www.wiley.com/en-ca/Chemometrics%3A+Data+Driven+Extraction+for+Science%2C+2nd+Edition-p-9781118904688.

[11] R.G. Brereton, Chemometrics for Pattern Recognition, John Wiley & Sons, Ltd, 2009. https://doi.org/10.1002/9780470746462.fmatter

[12] S. Brown, B. Walczak, R. Tauler, eds., Comprehensive Chemometrics, Segunda edición, Elsevier, Ámsterdam, 2020. https://www.sciencedirect.com/referencework/9780444641663/comprehensive-chemometrics.

[13] Y.S. Abu-Mostafa, M. Magdon-Ismail, H.-T. Lin, Learning From Data, AMLBook, 2012. https://work.caltech.edu/telecourse.