MATERIAL SUPLEMENTARIO

Big data analysis y machine learning en medicina intensiva

A continuación, hacemos una breve reseña del software más utilizado:

Lenguajes de programación

R(48): es un lenguaje de programación de código abierto especialmente diseñado para labores estadísticas y de aprendizaje máquina. Existen entornos de programación gratuitos en este lenguaje, como RStudio, que cuenta con interfaces de usuario funcionales e intuitivas que facilitan mucho al programador las tareas de análisis de los datos y de construcción y prueba de los modelos.

Python(49): la alternativa al anterior (al menos es necesario dominar uno de los dos para poder trabajar en este campo). Las bibliotecas de código abierto para ML se han desarrollado tradicionalmente en este lenguaje, aunque hoy en día hay versiones para otros entornos como R,Java o C++. Su aprendizaje es algo más complejo, pero a cambio es un lenguaje más potente y eficiente.

SQL,Structured Query Language(50): se utiliza para la exploración y explotación de bases de datos relacionales, que es donde se suele almacenar la información clínica estructurada de nuestros pacientes. Existen lenguajes equivalentes para realizar consultas en bases de datos no SQL como Cassandra o HBase, que permiten almacenar información no estructurada en un formato accesible para análisis.

Bibliotecas de código para análisis de datos (libraries, frameworks en inglés)

Scikit-learn(51): es la biblioteca de código más usada en ML, con implementación para los principales algoritmos, que pueden aplicarse de manera muy sencilla y eficiente a nuestros datos. En el repositorio de GitHub hemos incluido varios ejemplos que usan esta biblioteca, que está programada originariamente en Python, aunque existen versiones para otros lenguajes.

NLTK(52) y tidytext(53) son dos ejemplos de bibliotecas específicamente diseñadas para procesamiento de lenguaje natural.

Tensorflow(54) es la gran contribución de Google al aprendizaje máquina, y se utiliza fundamentalmente en el contexto de las redes neuronales y el “Deep learning”, aunque la biblioteca está siendo extendida a otras aplicaciones. Permite la utilización muy eficiente de hardware dedicado como son las GPU y TPU (chips especializados que permiten realizar muchas operaciones en paralelo), que aumentan en varios órdenes de magnitud la velocidad de obtención de resultados.

Spark MLib(55) y DeepLearning4j(56) son dos de las aportaciones de la comunidad de programadores Java al aprendizaje máquina. Aunque inicialmente Java se consideraba un lenguaje no adecuado por ser menos eficiente, hoy en día estas bibliotecas aportan una velocidad de procesamiento comparable a las que hasta ahora se consideraban más rápidas.

Bibliotecas de código auxiliares para extracción de datos, presentación gráfica, vectorización y otras: en este apartado se incluyen NumPy, SciPy, pandas, matplotlib y sciborn.

Table 1_Sup. Metodologías de aprendizaje máquina

Estrategia de aprendizaje	**Metodología**	**Resumen**
Supervisado	Perceptrón, Adaline(57)	Emplea “neuronas” artificiales que reciben como entradas los valores de las variables independientes.
Supervisado	Regresión logística(58)	Utiliza funciones lineales de las variables independientes con una función de activación sigmoide (logit), muy útil en clasificación de casos linealmente separables.
Supervisado	Máquina de vectores de soporte, Support vector machines (SVM)(58)	Extiende el algoritmo utilizado en el Perceptrón utilizando el concepto de la maximización de los márgenes de separación entre las distintas clases. Permite la clasificación de casos no linealmente separables.
Supervisado	Arboles de decisión, decision trees(58)	Utilizamos las variables independientes para crear un árbol de decisión que maximiza la ganancia de información en cada rama hasta llegar a un algoritmo de clasificación óptimo.
Supervisado	Bosques aleatorios(58), Random Forests, GBTs (Modelo de conjunto, Ensemble model)	Utiliza un conjunto de árboles de decisión y las técnicas de bootstrapping y de potenciación de gradiente para mejorar los resultados de los árboles de decisión individuales.

Supervisado	Potenciación adaptativa(59), Adaboost (modelo de conjunto, Ensemble Model)	Utiliza clasificadores simples que va combinando en cascada con una estrategia de aprendizaje por muestreo sin reemplazo.
Supervisado	Redes bayesianas(60)	Enfoque probabilístico que representa la interdependencia entre variables como grafos dirigidos acíclicos. Se usan por ejemplo para estimar la presencia de una enfermedad en función de una serie de síntomas.
Supervisado	Redes neuronales y aprendizaje profundo, “deep learning”(61)	Basado en las características operacionales de las redes neuronales biológicas, con capas de nodos que acumulan información y realizan inferencias que almacenan en sus interconexiones. Permiten clasificar eficazmente casos no linealmente separables. Útiles para explorar relaciones complejas entre las variables de entrada y salida del sistema.
Supervisado	Procesamiento de lenguaje natural(62)	Interpreta y analiza el lenguaje humano natural para ser procesado por un computador de manera eficiente, bien sea en codificación o clasificación de contenidos o para extracción de información relevante.
No supervisado	KNN, Valores cercanos, K-nearest neighbors(63)	Es una técnica de agrupación progresiva de registros para obtener una clasificación de estos en función de la proximidad de los valores de las variables.

No supervisado	Reglas de asociación(64, 65)	Saca a la luz relaciones interesantes entre las variables de grandes bases de datos.
No supervisado	Lógica inductiva(66)	Usa la programación funcional y lógica para derivar modelos de software que replican el conocimiento y las relaciones que existen en el sistema.
Supervisado o no supervisado	Aprendizaje de características o representación(67)	Técnicas para descubrir de manera autónoma el tipo de representación necesaria para la correcta detección de características de los datos para su clasificación a partir de los valores no tratados o “crudos”.
Por refuerzo	Monte Carlo, QLearning, SARSA(68)	El sistema debe perseguir un objetivo e ir aprendiendo a medida que se va explorando el entorno con el que interactúa, que puede no conocerse de antemano.
Metaheurística	Algoritmos genéticos(69)	Estrategias de optimización de soluciones aplicando a otros algoritmos mecanismos de selección natural, evolución, mutación y cruce, representando los parámetros del algoritmo como “genes” y las soluciones como “fenotipos”.