BLOQUE TEMÁTICO: INTELIGENCIA ARTIFICIAL Y CIENCIA DE DATOS

A1 (**)

70. Inteligencia artificial: Finalidad y clasificación: machine learning, deep learning, NLP, visión artificial, sistemas expertos, robótica, y agentes inteligentes. Aspectos éticos.

71. Ciencia de datos. Ciclo de vida de los procesos de modelado de datos (ETL, preprocesado, modelado, validación, MLops). Fundamentos estadísticos. Herramientas y lenguajes. Visualización de datos

A2 (**)

D5. Tipos abstractos de datos y estructuras de datos. Organizaciones de ficheros. Estrategias de diseño de algoritmos. Tipos de algoritmos: ordenación y búsqueda. Fundamentos de Inteligencia Artificial, tecnologías asociadas y áreas de aplicación.

BLOQUE TEMÁTICO: INTELIGENCIA ARTIFICIAL Y CIENCIA DE DATOS

6. MACHINE LEARNING

El aprendizaje automático o machine learning consiste en un conjunto de algoritmos que aprenden y resuelven problemas gracias a la experiencia. Hay diversos tipos de problemas que se abordan con técnicas de machine learning, entre ellos se encuentran los problemas de *clasificación* (donde queremos predecir una clase), los de *regresión*, las *series temporales*, etc.

Los algoritmos de aprendizaje automático pueden aprender de 4 formas distintas: mediante un **aprendizaje supervisado**, con **aprendizaje no supervisado**, con **aprendizaje semisupervisado** o con **aprendizaje por refuerzo**.

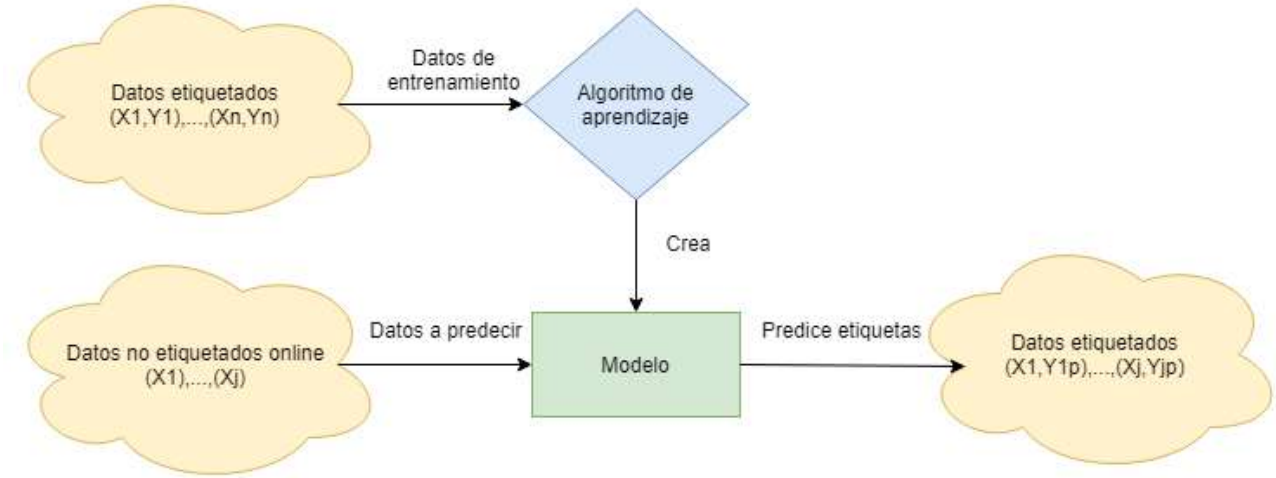
1. **Aprendizaje supervisado**: en este tipo de aprendizaje, el modelo se entrena utilizando datos etiquetados, es decir, datos que tienen una respuesta conocida. El objetivo es que el modelo aprenda a predecir la respuesta correcta para nuevos datos no etiquetados. Ejemplos de algoritmos de aprendizaje supervisado son la regresión lineal, los árboles de decisión y las redes neuronales

2. **Aprendizaje no supervisado**: en este tipo de aprendizaje, el modelo se entrena utilizando datos no etiquetados. El objetivo es que el modelo descubra patrones o estructuras ocultas en los datos. Los algoritmos de clustering y los algoritmos de reducción de dimensionalidad son ejemplos de algoritmos de aprendizaje no supervisado.

3. **Aprendizaje por refuerzo**: en este tipo de aprendizaje, el modelo aprende a través de la interacción con un entorno. El modelo recibe recompensas o castigos en función de sus acciones y aprende a tomar decisiones que maximicen las recompensas a largo plazo. Los algoritmos de aprendizaje por refuerzo se utilizan en aplicaciones como los juegos y la robótica.

Aprendizaje supervisado

Estos métodos son los más sencillos de realizar. En ellos se parte de un **conocimiento a priori**. El objetivo es, mediante unos datos de entrenamiento, deducir una función que haga lo mejor posible el mapeo entre unas entradas y una salida.



El aprendizaje supervisado es una técnica de machine learning que implica entrenar a una máquina para que aprenda a partir de un conjunto de datos etiquetados, es decir, datos que tienen una

respuesta conocida. El objetivo es que la máquina aprenda a predecir la respuesta correcta para nuevos datos no etiquetados. El aprendizaje supervisado se utiliza en aplicaciones como la **clasificación y la regresión**.

La **regresión lineal** es un algoritmo de aprendizaje supervisado que se utiliza para predecir valores numéricos continuos. La regresión lineal se basa en la relación lineal entre una variable independiente y una variable dependiente. El objetivo es encontrar la línea de mejor ajuste que minimice la distancia entre los puntos de datos y la línea. Una vez que se ha encontrado la línea de mejor ajuste, se puede utilizar para hacer predicciones sobre nuevos datos[2].

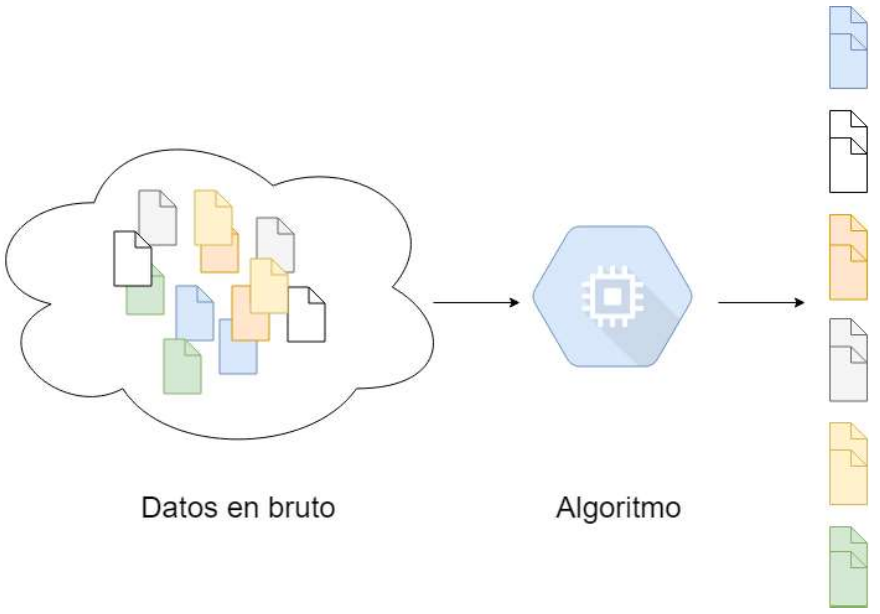
Los **árboles de decisión** son otro tipo de algoritmo de aprendizaje supervisado que se utiliza para la clasificación y la regresión. Los árboles de decisión se basan en la idea de dividir los datos en subconjuntos más pequeños y homogéneos en función de las características de los datos. Cada subconjunto se divide en subconjuntos más pequeños hasta que se alcanza un subconjunto que contiene solo datos de una clase o que tiene un valor de salida único. Una vez que se ha construido el árbol de decisión, se puede utilizar para hacer predicciones sobre nuevos datos[3].

Las **redes neuronales** son un tipo de algoritmo de aprendizaje supervisado que se utilizan en el aprendizaje profundo. Las redes neuronales se basan en la estructura del cerebro humano y están compuestas por capas de neuronas artificiales interconectadas. Cada neurona recibe entradas de otras neuronas y produce una salida que se utiliza como entrada para las neuronas de la siguiente capa. Las redes neuronales se utilizan en aplicaciones como el reconocimiento de imágenes, el procesamiento del lenguaje natural y la conducción autónoma

Aprendizaje no supervisado

Al contrario que en el aprendizaje supervisado, en este caso **no existe conocimiento a priori**.
El objetivo del aprendizaje no supervisado es modelizar la estructura o distribución de los datos para aprender más sobre ellos. Sirve tanto para entender como para resumir un conjunto de datos.

Se llama no supervisado porque, contrariamente al supervisado, tiende a ser **más subjetivo** ya que no tiene respuestas correctas. Los algoritmos sirven para descubrir y presentar estructuras interesantes en los datos.



En términos generales, pueden ser agrupados en algoritmos de clustering y algoritmos de asociación.

El aprendizaje no supervisado es una técnica de machine learning que se utiliza para descubrir patrones y estructuras ocultas en conjuntos de datos no etiquetados. A diferencia del aprendizaje supervisado, en el que los datos están etiquetados con una respuesta conocida, en el aprendizaje no supervisado no hay información previa sobre las categorías o clases a las que pertenecen los datos. Los algoritmos de aprendizaje no supervisado se utilizan principalmente en tareas de clustering y reducción de dimensionalidad.

Los algoritmos de **clustering** son una categoría de algoritmos de aprendizaje no supervisado que se utilizan para agrupar datos similares en conjuntos o clusters. El objetivo es encontrar grupos de datos que sean similares entre sí y diferentes de otros grupos. Los algoritmos de clustering se basan en medidas de similitud o distancia entre los datos y utilizan técnicas como k-means, clustering jerárquico y DBSCAN para agrupar los datos en función de su similitud.

Por otro lado, los algoritmos de **reducción de dimensionalidad** son otra categoría de algoritmos de aprendizaje no supervisado que se utilizan para reducir la cantidad de variables o características en un conjunto de datos. Estos algoritmos buscan encontrar una representación más compacta de los datos, manteniendo la mayor cantidad de información relevante posible. Algunos ejemplos de algoritmos de reducción de dimensionalidad son el Análisis de Componentes Principales (PCA) y el t-SNE (t-Distributed Stochastic Neighbor Embedding).

El clustering es ampliamente utilizado en diversas aplicaciones, como la segmentación de clientes, la agrupación de documentos, la detección de anomalías y la clasificación de imágenes. Por ejemplo, en el campo del marketing, el clustering se utiliza para identificar grupos de clientes con características similares, lo que permite personalizar las estrategias de marketing para cada grupo. En el campo de la medicina, el clustering se utiliza para agrupar pacientes con características similares, lo que puede ayudar en el diagnóstico y tratamiento de enfermedades.

La reducción de dimensionalidad es útil cuando se trabaja con conjuntos de datos de alta dimensionalidad, es decir, conjuntos de datos con un gran número de variables. Al reducir la dimensionalidad, se pueden eliminar variables redundantes o ruidosas, lo que puede mejorar la eficiencia y la precisión de los modelos de machine learning. Además, la reducción de dimensionalidad puede ayudar a visualizar los datos en espacios de menor dimensión, lo que facilita la interpretación y comprensión de los patrones presentes en los datos.

Aprendizaje semisupervisado

El aprendizaje semisupervisado se encuentra a medio camino entre el aprendizaje supervisado y el no supervisado.

Supone que los datos etiquetados y no etiquetados provienen de la **misma distribución**. Por otro lado, puede existir un sesgo en la elección de datos no etiquetados.

Entre los métodos de aprendizaje semisupervisado se encuentran:

- Selft-training
- Co-training
- Assemble
- Re-Weighting

Aprendizaje por refuerzo

El objetivo en el aprendizaje por refuerzo es aprender a mapear situaciones de acciones para maximizar una cierta función de recompensa. En estos problemas un agente aprende por prueba y error en un ambiente dinámico e incierto. En cada interacción el agente recibe como entrada un indicador de estado actual y selecciona una determinada acción que maximice una función de

refuerzo o recompensa a largo plazo. Este proceso de decisión secuencial se puede caracterizar como un proceso de Markov.

El aprendizaje por refuerzo es una técnica de aprendizaje automático en la que un agente aprende a través de la interacción con un entorno. El agente recibe recompensas o castigos en función de sus acciones y aprende a tomar decisiones que maximicen las recompensas a largo plazo. El aprendizaje por refuerzo se basa en el concepto de trial and error, donde el agente explora diferentes acciones y aprende de las consecuencias de esas acciones.

Existen varios algoritmos de aprendizaje por refuerzo que se utilizan en diferentes aplicaciones. A continuación, se presentan algunos ejemplos de estos algoritmos:

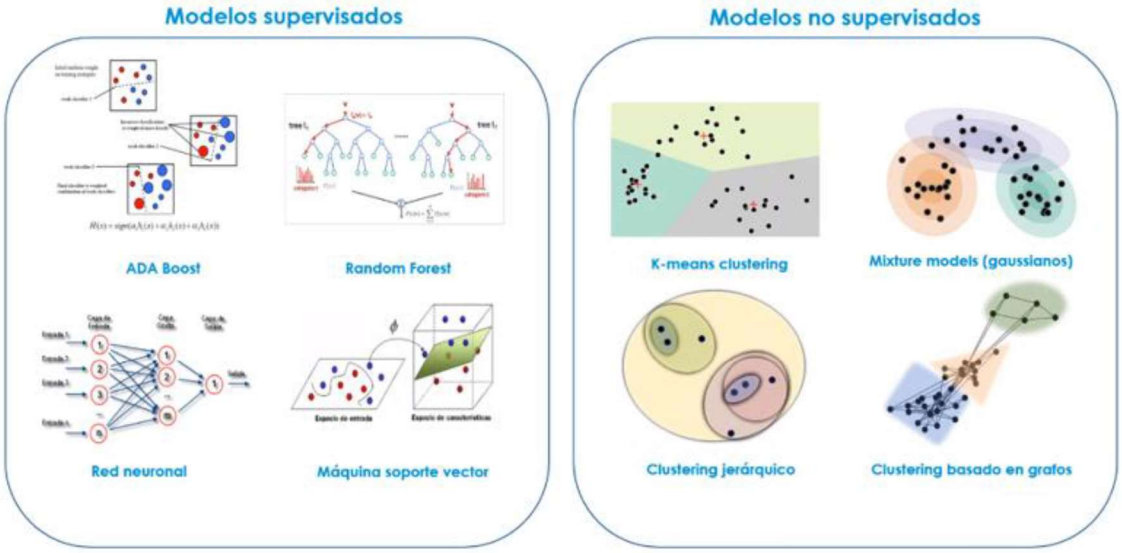
1. Q-Learning: es uno de los algoritmos más conocidos de aprendizaje por refuerzo. Se utiliza en entornos donde el agente toma decisiones secuenciales y aprende a través de la actualización de una función de valor llamada Q-Value. El algoritmo utiliza una tabla de valores Q para almacenar las estimaciones de recompensa esperada para cada par estado-acción. A medida que el agente interactúa con el entorno, actualiza los valores Q en función de las recompensas recibidas y utiliza estos valores para tomar decisiones futuras.

2. SARSA: es otro algoritmo popular de aprendizaje por refuerzo. Al igual que Q-Learning, SARSA se utiliza en entornos secuenciales y actualiza los valores de Q-Value. Sin embargo, a diferencia de Q-Learning, SARSA utiliza una política de toma de decisiones basada en la acción actual y la siguiente acción (State-Action-Reward-State-Action). Esto significa que el agente aprende a través de la experiencia real de tomar una acción y observar la siguiente acción y la recompensa asociada.

3. Algoritmos genéticos: estos algoritmos se inspiran en la teoría de la evolución y se utilizan en problemas de optimización. En el aprendizaje por refuerzo, los algoritmos genéticos se utilizan para encontrar la mejor combinación de acciones que maximice la recompensa acumulada. El algoritmo crea una población de soluciones candidatas y las evalúa en función de su rendimiento. Luego, se seleccionan las mejores soluciones y se aplican operadores genéticos como la reproducción y la mutación para generar nuevas soluciones. Este proceso se repite hasta encontrar la solución óptima.

4. A3C (Asynchronous Advantage Actor-Critic): es un algoritmo de aprendizaje por refuerzo que combina elementos de aprendizaje supervisado y no supervisado. Utiliza una red neuronal para estimar los valores de recompensa y una política para tomar decisiones. A3C utiliza múltiples agentes que interactúan con el entorno de forma asíncrona y comparten sus experiencias para mejorar el rendimiento del aprendizaje. Este enfoque permite una mayor eficiencia computacional y una mejor exploración del espacio de acciones.

Estos son solo algunos ejemplos de algoritmos de aprendizaje por refuerzo. Cada algoritmo tiene sus propias características y se utiliza en diferentes contextos y aplicaciones. El aprendizaje por refuerzo se utiliza en una amplia gama de áreas, como la robótica, los juegos, la optimización de recursos y la toma de decisiones en entornos complejos y cambiantes. Es un campo de investigación activo y en constante evolución, con el objetivo de desarrollar algoritmos más eficientes y efectivos para el aprendizaje de agentes inteligentes.



NOTA: Algoritmo "Random Forest"

El algoritmo random forest se basa en la idea de combinar varios árboles de decisión para obtener una predicción más precisa y robusta que la de un solo árbol. Un árbol de decisión es una estructura que divide los datos en subconjuntos según unas reglas basadas en las características de los datos. Por ejemplo, si queremos clasificar animales según su especie, podemos usar un árbol de decisión que pregunte si el animal tiene pelo, si es carnívoro, si tiene alas, etc. Cada pregunta divide los datos en dos ramas, hasta llegar a las hojas del árbol, donde se asigna una clase a cada subconjunto.

Un árbol de decisión tiene la ventaja de ser fácil de interpretar y de manejar tanto problemas de clasificación como de regresión. Sin embargo, también tiene algunos inconvenientes, como que puede sobreajustarse a los datos de entrenamiento y perder capacidad de generalización, o que puede ser muy sensible a pequeños cambios en los datos o en las reglas de división.

Para solucionar estos problemas, el algoritmo random forest propone construir muchos árboles de decisión diferentes usando dos técnicas: el bagging y la selección aleatoria de características.

El bagging consiste en generar subconjuntos de datos a partir del conjunto original mediante un muestreo con reemplazo. Es decir, se eligen al azar tantos datos como tenga el conjunto original, pero se permite que algunos se repitan y otros se queden fuera. De esta forma, cada subconjunto es diferente y representa una variación del conjunto original. Luego, se entrena un árbol de decisión con cada subconjunto, lo que hace que cada árbol sea diferente y tenga menos varianza que uno solo.

La selección aleatoria de características consiste en elegir al azar un número menor de características que las disponibles para construir cada árbol. Esto hace que cada árbol sea más independiente de los demás y tenga menos correlación entre ellos. Además, reduce la complejidad computacional y evita que algunas características dominen el proceso de división.

Una vez construidos los árboles, el algoritmo random forest combina sus predicciones mediante una regla de agregación. Para problemas de clasificación, se usa la regla de la mayoría: se elige la clase que más árboles hayan predicho. Para problemas de regresión, se usa la media: se calcula el promedio de los valores predichos por los árboles.

7. DEEP LEARNING

El Deep Learning, o aprendizaje profundo, es una técnica de aprendizaje automático basada en el modelo de red neuronal artificial. Consiste en apilar decenas o incluso cientos de capas de neuronas para aportar mayor complejidad al establecimiento de reglas y patrones en los datos. El objetivo del Deep Learning es imitar las acciones del cerebro humano mediante estas redes neuronales artificiales.

Las redes neuronales artificiales, que son la base del Deep Learning, tratan de imitar el cerebro humano a través de una combinación de entradas de datos, conexiones ponderadas y funciones de activación. Estas redes están compuestas por múltiples capas, incluyendo una capa de entrada, capas ocultas y una capa de salida. Cada capa está formada por un conjunto de neuronas interconectadas, y la información fluye a través de estas conexiones para realizar cálculos y tomar decisiones.

Existen diferentes tipos de Deep Learning, cada uno con sus propias características y aplicaciones específicas. Algunos ejemplos son:

1. Redes Neuronales Convolucionales (CNN): son ampliamente utilizadas en tareas de visión por computadora, como el reconocimiento de imágenes y el procesamiento de video. Las CNN son capaces de extraer características relevantes de las imágenes mediante la aplicación de filtros convolucionales y la reducción de la dimensionalidad de los datos.

2. Redes Neuronales Recurrentes (RNN): son adecuadas para el procesamiento de secuencias y datos secuenciales, como el procesamiento del lenguaje natural y la generación de texto. Las RNN utilizan conexiones recurrentes que permiten a la red tener memoria y capturar dependencias a largo plazo en los datos.

3. Redes Generativas Adversariales (GAN): son utilizadas para generar datos sintéticos realistas, como imágenes, música o texto. Las GAN consisten en dos redes neuronales: un generador que crea datos sintéticos y un discriminador que intenta distinguir entre los datos reales y los sintéticos. Estas dos redes se entrenan de forma adversarial, mejorando constantemente su rendimiento.

8. HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES DE MACHINE LEARNING Y DEEP LEARNING

Existen varios lenguajes de programación que se pueden utilizar para desarrollar herramientas de machine learning y deep learning. A continuación, se presentan algunos de los más comunes:

1. Python: Python es uno de los lenguajes de programación más populares para el desarrollo de herramientas de machine learning y deep learning. Es fácil de aprender, tiene una sintaxis clara y concisa, y cuenta con un amplio ecosistema de herramientas y librerías para análisis de datos y visualización.

2. Java: Java es otro lenguaje de programación popular para el desarrollo de herramientas de machine learning y deep learning. Es un lenguaje de programación orientado a objetos y se utiliza ampliamente en aplicaciones empresariales.

3. C++: C++ es un lenguaje de programación de alto rendimiento que se utiliza en aplicaciones que requieren un procesamiento rápido de grandes cantidades de datos. Es popular en el desarrollo de herramientas de machine learning y deep learning que requieren una gran cantidad de cálculos matemáticos.

4. R: R es un lenguaje de programación especializado en análisis estadístico y visualización de datos. Es popular en el desarrollo de herramientas de machine learning y deep learning que requieren un análisis estadístico detallado.

Algunas de las librerías de Python más utilizadas para el desarrollo de herramientas de machine learning y deep learning:

1. Pandas: Una librería de manipulación y análisis de datos que proporciona estructuras de datos flexibles y eficientes para trabajar con conjuntos de datos.

2. Numpy: Una librería fundamental para el cálculo numérico en Python. Proporciona una estructura de datos de matriz multidimensional y funciones para realizar operaciones matemáticas en ellas.

3. SciPy: Una librería que proporciona funcionalidades avanzadas para la computación científica, incluyendo algoritmos de optimización, procesamiento de señales, álgebra lineal y más.

4. Scikit-learn: Una librería de aprendizaje automático de propósito general que proporciona una amplia gama de algoritmos y herramientas para tareas como clasificación, regresión, agrupamiento y selección de características.

5. Matplotlib: Una librería de visualización de datos que permite crear gráficos y visualizaciones de alta calidad en Python.

6. TensorFlow: Una librería de código abierto desarrollada por Google para el desarrollo de modelos de aprendizaje automático y deep learning. Proporciona una plataforma flexible para la construcción y entrenamiento de redes neuronales.

7. Keras: Una librería de alto nivel que se ejecuta sobre TensorFlow y simplifica el proceso de construcción y entrenamiento de redes neuronales.

Las librerías de Python para machine learning y deep learning tienen algunas diferencias. A continuación, se presentan algunas de las diferencias más relevantes: