Informática y comunicaciones

ies azarquiel

Curso de especialización inteligencia artificial y big data

Tema 1: Introducción al Aprendizaje Automático

Profesor: Sebastián Rubio

Septiembre 2025

Sistema aprendizaje automático

1 Inteligencia Artificial Débil

1.1 Características Fundamentales de la IA Débil

Se refiere la IA que se utiliza para resolver un problema específico. Casi todas la aplicaciones de inteligencia artificial que tenemos hoy en día son de este tipo.

- Enfoque en tareas específicas.
- Carencia de conciencia, subjetividad o autoconciencia.
- Dependencia de grandes cantidades de datos y entrenamiento.
- Explicación con ejemplos concretos (reconocimiento de voz, sistemas de recomendación, filtros de spam, etc.).

1.2 Aplicaciones de la IA Débil

- **Aprendizaje Automático (Machine Learning):**
  - Clasificación, regresión, clustering.
  - Ejemplos: diagnóstico médico, detección de fraude, predicción de demanda.
- **Procesamiento del Lenguaje Natural (NLP):**
  - Traducción automática, chatbots, análisis de sentimiento.
  - Ejemplos: asistentes virtuales, herramientas de resumen de texto.
- **Visión por Computador (Computer Vision):**
  - Reconocimiento de objetos, detección facial, análisis de imágenes médicas.
  - Ejemplos: coches autónomos, sistemas de vigilancia.
- **Robótica Inteligente:**
  - Automatización industrial, robots de servicio.
  - Ejemplos: líneas de ensamblaje, aspiradoras inteligentes.

1.3 Ventajas de la IA Débil

- Automatización de tareas repetitivas y tediosas.
- Mejora de la eficiencia y la productividad.
- Capacidad para analizar grandes volúmenes de datos rápidamente.
- Reducción de errores humanos.
- Nuevas posibilidades en diversos campos.

1.4 Inconvenientes y Desafíos de la IA Débil

- Falta de flexibilidad y adaptabilidad a tareas no programadas.
- Dependencia de la calidad y cantidad de los datos de entrenamiento.
- Posibilidad de sesgos en los datos que llevan a resultados injustos.
- Impacto en el empleo en ciertos sectores.
- Cuestiones éticas relacionadas con la privacidad y la seguridad de los datos.

1.5 Usos y Posibilidades Futuras de la IA Débil

- Integración en dispositivos cotidianos (IoT).
- Personalización de servicios y experiencias.
- Avances en la medicina y la investigación científica.
- Soluciones para problemas complejos en áreas como el cambio climático.
- **Debate:** ¿Qué nuevas aplicaciones de la IA débil podemos imaginar?

2 Inteligencia Artificial Fuerte

Es un tipo I.A que sirve para resolver problemas generales. Es como un ser humano que es capaz de aprender,pensar, inventar y resolver problemas más complicados. La singularidad es una I.A que supera a la inteligencia humana.

2.1 Características Definitorias de la IA Fuerte

- Capacidad de comprender, aprender y aplicar conocimiento en una amplia gama de tareas, similar a la inteligencia humana.
- Potencial para la conciencia, la subjetividad y la autoconciencia (aspectos aún en debate).
- Habilidad para razonar, planificar, resolver problemas complejos y pensar de forma abstracta.
- Diferenciación entre la IA fuerte y la Inteligencia Artificial General (IAG).

2.2 Aplicaciones Potenciales de la IA Fuerte (Escenarios Teóricos)

- Resolución de problemas globales complejos (cambio climático, pandemias).
- Investigación científica avanzada y descubrimiento de nuevos conocimientos.
- Exploración espacial y colonización de otros planetas.
- Creación de nuevas formas de arte y cultura.
- Interacción humano-máquina a un nivel completamente nuevo.

2.3 Ventajas Potenciales de la IA Fuerte

- Superación de las limitaciones de la inteligencia humana en ciertos aspectos.
- Innovación y progreso a un ritmo sin precedentes.
- Soluciones a problemas que actualmente son inabordables.
- Potencial para mejorar significativamente la calidad de vida humana.

2.4 Inconvenientes y Riesgos Potenciales de la IA Fuerte

- Desafíos éticos y morales sin precedentes (derechos de las máquinas, control).
- Riesgo de pérdida de control si la IA fuerte no está alineada con los valores humanos.
- Impacto socioeconómico masivo, incluyendo la posible obsolescencia del trabajo humano.

- Escenarios de "singularidad tecnológica" y sus implicaciones.
- **Debate:** ¿Cómo podemos mitigar los riesgos de la IA fuerte?

2.5 Usos y Posibilidades Futuras de la IA Fuerte (Consideraciones a Largo Plazo)

- La búsqueda de la IAG como un objetivo a largo plazo.
- Diferentes enfoques y arquitecturas para intentar alcanzar la IA fuerte.
- La importancia de la investigación en ética y seguridad de la IA.
- Reflexión sobre el futuro de la humanidad en un mundo con IA fuerte.

El aprendizaje automático es un subconjunto de técnicas de inteligencia artificial. Esta a su vez se puede clasificar en supervisado, no supervisado y por refuerzo. En este módulo se estudia técnicas de aprendizaje supervisado y no supervisado

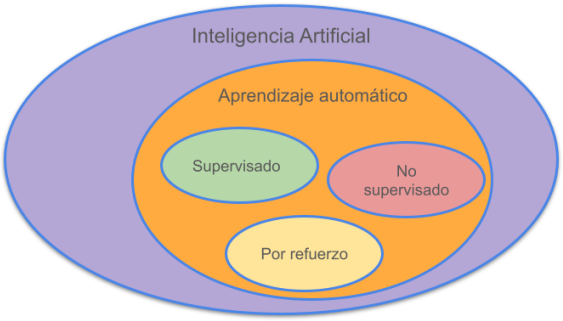

Figure 1: Esquema 1

3 INTRODUCCIÓN AL APRENDIZAJE AUTOMÁTICO

Cuando decimos que las máquinas 'aprenden', lo hacen mediante algoritmos matemáticos: nosotros proporcionamos datos de entrada y salida, y se genera un modelo que, ajustándose a los datos, es capaz de generar una salida correcta si le proporcionamos una entrada nueva similar a la anteriores. Lo que quiere es que las máquinas aprendan sin ser programas con reglas específicas. Se aplican métodos matemáticos para detectar patrones en los datos y, a partir de ahí, hacer predicciones e incluso tomas

decisiones.

Los algoritmos de aprendizaje automático existen desde hace varias décadas, pero el desarrollo de la tecnología, el incremento del poder de cálculo y de almacenamiento de datos, ha hecho posible su desarrollo y presencia a nivel universal. Su uso va desde lectura compresiva, traducción, reconocimiento de imágenes. En ciertas actividades superan a las capacidades del ser humano, sobre todo en tareas que son muy repetitivas. No es un aprendizaje de inferencia como el humano, pero a mayor número de datos procesado sí se produce una cierta mejora.

Como cualquier programa informático necesita de un humano que lo programe

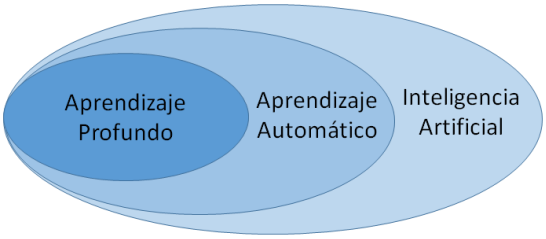

Figure 2: Esquema 2

y supervise, éste es uno de los trabajos del científico de datos. Por ejemplo para que un modelo reconozca la imagen de un gato sin necesidad de programar, será necesario que el algoritmo pertinente haya analizado un número grande de imágenes de gatos, de modo que luego sea capaz que una imagen cualquiera detecte la presencia de un gato. El científico de datos elige qué algoritmo es el más adecuado para resolver el problema con los datos disponibles y de configurarlo matemáticamente, ajustando parámetros y minimizando funciones de error, para procesar mejor los datos y con mejores resultados
Dentro del aprendizaje automático hay otra disciplina conocida como aprendizaje profundo, esta se apoya principalmente en el concepto de redes neuronales. Un red neuronal no es nada más que un artilugio matemático al que se pasan datos y obtenemos unas salidas. La red neuronal más simple consta de tres capas: capa de entrada, capa de procesamiento o capa oculta, y capa de salida. Una red neuronal aprende mediante el ajuste automático de sus parámetros hasta que los resultados sean correctos.

4 ANÁLISIS DE DATOS. ETAPAS

Para entrenar un modelo tenemos que partir de un conjunto de datos. identificando cuáles son la variables de entrada y cuales serán las de salida, si la hubiese. En la siguiente imagen se puede ver de forma esquemática las diferentes etapas

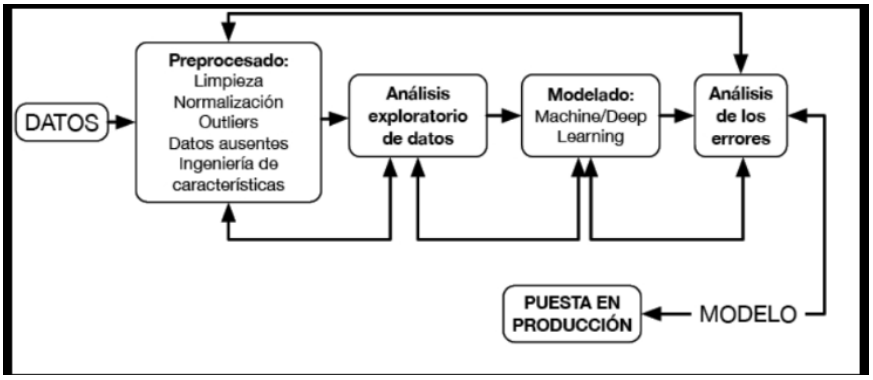

Figure 3: Esquema 2

4.1 Preprocesado

Esta fase es la que más tiempo requiere para entrenar correctamente un modelo, consta de las siguientes tareas

- Limpieza de datos, en el conjunto de datos no pueden haber valores ausentes, datos incorrectos y el rango de valores sea el correcto (outliers).
- Normalización, la diferencia de rangos entre variables puede impactar negativamente el resultado de algunos modelos machine learning, por ejemplo, si tenemos una variable que tome valores entre 0 y 106 y otra que tome valores en 0 y 0,0003, la mayoría de los algoritmos dará más importancia a la primera variable, por lo tanto se hace necesario que las variables que son importantes en el entrenamiento estén normalizadas.
- Detección de outliers, un valor outlier es aquel que destaca sobre los demás, es decir se sale del patrón general de valores, esto puede ser debido, en ocasiones, a la introducción errónea del dato, a la grabación incorrecta, etc.
- Datos ausentes, en una determinada variable puede faltar un valor. Si se trata de una variable discreta podemos sustituirlo por la moda (valor que más se repite). Si la variable es contínua se suele sustituir por la media o mediana.

Hay que tener encuenta que completando de esta forma los valores restantes estamos 'modificando' la realidad, afectando de esta forma a los resultado que arroja el modelo

- Ingenieria de características, en esta etapa lo que se hace *crear/eliminar variable*. Otra tarea es la de *seleccionar/descartar variable*

4.2 Análisis exploratorio de datos

Es el momento de captar la información que contienen nuestros datos. En las variables contínuas se hace mediante el cálculo de los estadísticos descriptivos más comunes, y en las variables discretas. Se evalúa mediante contrastes de hipótesis (estadística inferencial), la posible existencia de relaciones, o en caso contrario, la independencia entre las variables y su normalidad.

4.3 Modelado

Es el modelo que se desarrolla, aprendizaje automáticao o aprendizaje profundo, esto va a depender de la cantidad de datos y del problema a resolver. Y cómo no existe ningún modelo perfecto, habrá que probar con varias hasta dar con el que mejor se ajuste. En cualquier caso debemos de evitar el sobreajuste del modelo, estos memorizan los datos y sus resultados, y ante nuevos datos siempre se obtiene resultados erróneos.

4.4 Análisis de los errores

Una vez que hemos aprobado nuestro modelo, es decir aquel que minimiza el error y tiene la mayor capacidad de generalización. Ahora toca validarlo mediante diferentes medidas de error que además, podemos usar para comparar con otros modelos y elegir, finalmente, el que mejor se adecue a nuestro problema y datos. El sesgo es la diferencia entre el valor medio predicho por el modelo y el valor real, Un alto sesgo da lugar a modelos subajustados (underfitting) por lo que el sesgo debe ser bajo para asegurar que el modelo se ajusta a la realidad.
La *varianza*mide la tendencia a aprender cosas irrelevantes o muy peculiares para el objetivo. Una varianza alta significa que el modelo esta sobreajustado (overfitting) por lo que debe ser baja para asegurar que e modelo tiene la capacidad de generalizar.

4.5 Puesta en producción

Una vez que el modelo ha pasado todas las etapas previas, deberemos revisar varios puntos antes de que se pueda utilizar de forma masiva en el entorno de la indus-

tria/empresa o sector público.

- Proporcionar una excelente experiencia de usuario
- Evaluar la seguridad del acceso al modelo
- Identificar la capacidad del modelo de explicabilidd, imparcialidad, confiabilidad y privacidad
- Examinar la velocidad de actualización y de respuesta del modelo

5 ALGORITMOS DE APRENDIZAJE AUTOMÁTICO

En el aprendizaje automático (machine learning), se distinguen dos técnicas en función de los objetivos que se persiguen al crear un modelo. Por un lado esta algoritmo supervisados (etiquetados), son aquellos en los que conocemos los posibles resultados que se pueden obtener, y por otro los algoritmos no supervisados (no etiquetados), son aquellos donde dejamos que sea el propio algoritmo el que decida que posibles soluciones hay.

5.1 Aprendizaje supervisado

En el aprendizaje supervisado se utiliza un conjunto de datos etiquetados para entrenar el algoritmo. Los datos etiquetados son aquellos que han sido "marcados" con las respuestas correctas bajo supervisión humana para evitar errores en el modelo. El proceso de etiquetado es caro y laborioso. Las etiquetas representan los posibles valores que nos podemos encontrar. Por ejemplo, si queremos construir un sistema de reconocimiento de imágenes de animales, entrenaríamos el algoritmo con dichas imágenes previamente etiquetadas y le pediríamos que reconociera los animales en imágenes nuevas. El modelo que se obtenga deberá devolver como resultado una de las etiquetas seleccionadas. A continuación se dá un listado ordenado por dificultad de los algoritmos supervisados. **aprendizaje supervisado** según su complejidad conceptual y técnica, desde los más simples hasta los más avanzados.

5.1.1 Nivel Básico

En este nivel se encuentran los algoritmos más sencillos de entender e implementar, que sirven como punto de partida y línea base para cualquier proyecto.

- **Regresión Lineal**: Es el pilar absoluto. Su objetivo es simple: modelar la relación entre una variable dependiente y una o más variables independientes

ajustando una línea recta (o un hiperplano). La intuición detrás de "minimizar la suma de los errores al cuadrado" es geométricamente clara y accesible.

- **Regresión Logística**: A pesar de su nombre, es un algoritmo de clasificación lineal. Es el siguiente paso natural después de la regresión lineal. Su concepto de modelar la probabilidad de pertenencia a una clase usando una función sigmoide es muy intuitivo. Es el algoritmo de referencia para problemas de clasificación binaria.

5.1.2 Nivel Intermedio

Estos algoritmos imitan la toma de decisiones humana y son muy fáciles de interpretar.

- **Árboles de Decisión** (*Decision Trees*:Su lógica es supremamente intuitiva: una serie de preguntas de sí/no (if-else) que llevan a una decisión final. La facilidad para visualizar y explicar el modelo lo coloca en un nivel de dificultad muy bajo. La complejidad comienza a asomarse al elegir la profundidad del árbol y la métrica para dividir los nodos (como la impureza de Gini o la ganancia de información), pero su concepto central es sencillo.

- **Máquinas de Vectores de Soporte (SVM - Support Vector Machines)**: Para problemas lineales, su concepto de encontrar el "mayor margen" entre clases es comprensible. Sin embargo, su dificultad da un salto significativo cuando utilizamos el "kernel trick" para problemas no lineales. Entender cómo se proyectan los datos a espacios de n dimensiones (hiperdimensionales) para hacerlos separables requiere una base sólida en álgebra lineal y hace que las SVM sean considerablemente más complejas que los modelos lineales básicos.

- **Algoritmos de Ensamble (Ensemble Methods)**: Estas técnicas se basan en la sabiduría de las multitudes, combinando múltiples modelos débiles para formar uno fuerte y robusto.

  - **Random Forest**: Es un conjunto de muchos Árboles de Decisión. Su idea de promediar las predicciones de muchos árboles (bagging) para reducir el sobreajuste es un concepto de dificultad media que se apoya en la simplicidad de los árboles base.
  - **Gradient Boosting (XGBoost, LightGBM, CatBoost)**: Representan un paso de dificultad superior a Random Forest. En lugar de construir árboles de forma independiente, los construye de forma secuencial, donde cada nuevo árbol intenta corregir los errores del anterior. Este concepto de "aprender de los errores pasados" y la optimización mediante el descenso de

gradiente lo convierten en uno de los algoritmos más potentes pero también más complejos de entender en profundidad y de ajustar correctamente.

5.1.3 Nivel Avanzado

Este nivel representa la vanguardia del aprendizaje supervisado, con modelos extremadamente flexibles pero complejos.

- **Redes Neuronales Artificiales (ANN) y Perceptrón Multicapa (MLP)**: La complejidad aquí es notoria. Se requiere comprender la arquitectura de capas (entrada, ocultas, salida), las funciones de activación, la propagación hacia adelante y, lo más crucial, el algoritmo de retropropagación (backpropagation) para ajustar los pesos mediante el descenso de gradiente. La cantidad de hiperparámetros a configurar (capas, neuronas, tasa de aprendizaje, etc.) y la necesidad de poder computacional los colocan en un escalón de alta dificultad.

5.1.4 Nivel Experto

La cumbre de la complejidad en el aprendizaje supervisado, reservedada para datos con estructuras inherentemente complejas.

- **Redes Neuronales Convolucionales (CNN)**: Especializadas en datos de imagen. Su dificultad radica en entender los componentes especializados que las forman: convoluciones (filtros que detectan características), pooling (para reducir la dimensionalidad) y capas totalmente conectadas. El concepto de que las primeras capas detecten bordes y texturas, y las capas profundas detecten objetos complejos, añade una capa de abstracción significativa.

- **Redes Neuronales Recurrentes (RNN) y LSTM/GRU**: Diseñadas para datos secuenciales (series de tiempo, texto). Son notablemente más difíciles que las redes feedforward debido a su "memoria". Comprender cómo la información persiste a través del tiempo, y en particular el mecanismo de las Long Short-Term Memory (LSTM) networks con sus puertas (forget, input, output) para resolver el problema del gradiente vanishing, es considerado uno de los temas más desafiantes en el aprendizaje automático.

5.2 No Supervisado

El aprendizaje no supervisado es la rama del aprendizaje máquina que más cerca está del aprendizaje biológico junto con el aprendizaje reforzado. Son sistemas que no necesitan una señal que les indique si lo están haciendo bien o mal, como son

los sistemas de aprendizaje supervisado o aprendizaje reforzado. Mientras que en un aprendizaje supervisado el objetivo es establecer un mapeo entre dos conjuntos de datos (normalmente especificado como X e Y), en un aprendizaje no supervisado no existe ese segundo conjunto de datos. El objetivo se centra en explorar y obtener posibles patrones, repeticiones o estructuras en los datos que se tienen.

Figure 4: Enter Caption

5.2.1 Nivel básico

En este escalón encontramos los algoritmos más fundamentales e intuitivos.

- **K-Means**: Es el algoritmo de agrupamiento por excelencia. Su concepto es simple: encontrar un número K de grupos (clusters) representados por sus centroides. La dificultad radica principalmente en seleccionar la K correcta y en su sensibilidad a los valores atípicos (outliers) y a la inicialización aleatoria. Es la puerta de entrada al clustering.

5.2.2 Nivel Intermedio

- **Agrupamiento Jerárquico (Hierarchical Clustering)**: Su idea de construir un árbol de clusters (dendrograma) es muy visual y intuitiva. La complejidad aumenta ligeramente al tener que elegir la métrica de distancia adecuada y el criterio de enlace (linkage) entre clusters, además de ser más costoso computacionalmente que K-Means.

- **PCA (Análisis de Componentes Principales)**: Es el pilar de la reducción de dimensionalidad. Su objetivo geométrico (encontrar las direcciones de máxima varianza) es fácil de entender. La barrera para algunos puede estar en la base matemática, que involucra autovectores y autovalores, pero a un nivel práctico su implementación es muy straightforward.

- **Apriori**: El algoritmo clásico para la minería de reglas de asociación (como el famoso análisis de la cesta de la compra). Su lógica de "itemsets frecuentes"

es clara, pero requiere comprender bien las métricas de soporte, confianza y lift para filtrar las reglas útiles.

- **DBSCAN (Density-Based Spatial Clustering)**:Introduce el poderoso concepto de agrupar por densidad en lugar de por proximidad a un centroide. Esto lo hace robusto a outliers y capaz de encontrar clusters de formas arbitrarias. La dificultad principal reside en configurar eficazmente sus dos parámetros clave, lo cual puede no ser trivial

- **GMM (Modelos de Mezcla Gaussianas)**: Un paso adelante en sofisticación respecto a K-Means. En lugar de centroides, modela los clusters como distribuciones de probabilidad gaussianas, lo que permite clusters de forma elíptica. La comprensión del algoritmo de Expectación-Maximización (EM) que utiliza detrás eleva su dificultad conceptual.

5.2.3 Nivel Avanzado

Aquí nos adentramos en algoritmos con una base matemática más compleja o con hiperparámetros muy sensibles.

- **t-SNE (t-Distributed Stochastic Neighbor Embedding)**: Una herramienta magnífica para la visualización de datos de alta dimensión en 2D o 3D. Su dificultad no está en usarlo, sino en entender cómo funciona (minimiza la divergencia de Kullback-Leibler) y en interpretar correctamente sus resultados, ya que es estocástico (los resultados varían) y sus parámetros como la perplejidad son delicados.

- **UMAP (Uniform Manifold Approximation and Projection)**: Surgió como una alternativa más moderna y rápida a t-SNE, preservando mejor la estructura global de los datos. Se basa en principios de topología algebraica (reconstrucción de variedades), lo que lo hace conceptualmente complejo, aunque su uso práctico sea similar al de t-SNE.

5.2.4 Nivel Experto

Este es el nivel de mayor complejidad, que requiere conocimientos de redes neuronales y deep learning

- **SOM (Mapas Auto-Organizados o Mapas de Kohonen)**: Son un tipo de red neuronal artificial no supervisada utilizada para reducir dimensionalidad y visualizar clusters. La dificultad surge de la combinación de conceptos: la arquitectura de la red (una malla neuronal), el aprendizaje competitivo (la "neurona ganadora") y la adaptación topológica.

- **Autoencoders (Codificadores Automáticos)**: Representan la vanguardia en reducción de dimensionalidad no lineal y extracción de características. Son redes neuronales que aprenden a comprimir (codificar) y reconstruir (decodificar) los datos. Su alta dificultad proviene de la necesidad de diseñar la arquitectura de la red, elegir funciones de activación y pérdida, y ajustar los hiperparámetros típicos del deep learning, además de requerir un mayor poder computacional.

Los criterios de clasificación seguidos son:

- **Interpretabilidad**: Facilidad de explicación.
- **Hiperparámetros**: Cantidad de ajustes necesarios.
- **Matemáticas Involucradas**: Complejidad teórica.
- **Capacidad de Generalización**: Necesidad de datos y fine-tuning.

Este orden es una guía general. La dificultad percibida puede variar enormemente según tu formación. Un matemático podría encontrar las SVM más sencillas que los árboles, mientras que un programador podría entender Random Forest más rápido que la regresión logística. La clave es dominar los fundamentos (Regresión Lineal/Logística) antes de escalar a la complejidad de los ensembles y las redes neuronales. El algoritmo más complejo no es siempre la mejor solución;a menudo, la simplicidad y la interpretabilidad ganan.