

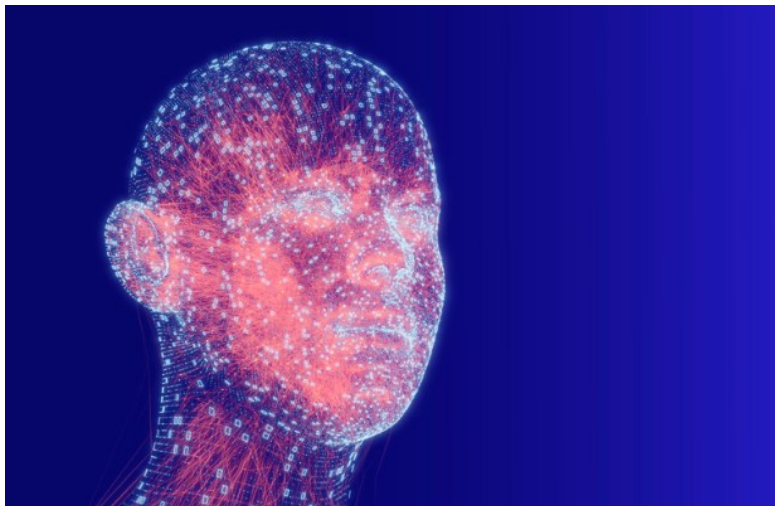
Auto Machine Learning

1. Inmersión al Auto Machine Learning y a las plataformas más relevantes.

Auto Machine Learning hace referencia al conjunto de herramientas que automatizan partes del ciclo de creación de modelos de Machine Learning. En un proceso tradicional, una persona debe probar diferentes algoritmos, preparar transformaciones, ajustar parámetros y comparar métricas. En AutoML, la plataforma realiza muchas de esas pruebas de forma automática y presenta una selección de modelos candidatos.

Por ejemplo, si una empresa quiere predecir el abandono de clientes, puede cargar un conjunto de datos con variables como antigüedad, uso del servicio, reclamaciones, facturación y estado final del cliente. Una herramienta AutoML puede analizar el dataset, identificar el tipo de problema, probar varios modelos de clasificación, ajustar configuraciones y mostrar cuál obtiene mejores resultados.

Entre las plataformas más relevantes se encuentran AWS SageMaker Autopilot, Google Vertex AI AutoML, Azure Machine Learning AutoML, H2O AutoML, Auto-sklearn, TPOT y DataRobot. Algunas están orientadas a entornos empresariales en la nube y otras pueden utilizarse en proyectos locales o experimentales.



AWS SageMaker Autopilot permite automatizar el desarrollo de modelos y puede generar artefactos del modelo y endpoints para exponer predicciones mediante API, según la documentación oficial de AWS.

La elección de una plataforma depende del contexto. Si una empresa ya trabaja con AWS, SageMaker puede encajar bien. Si se busca una opción local o de código abierto, H2O AutoML, Auto-sklearn o

TPOT pueden ser alternativas interesantes. Si se necesita integración empresarial, trazabilidad, despliegue y monitorización, las plataformas cloud suelen ofrecer más servicios conectados.

2. Introducción al Auto ML y a los beneficios que puede proporcionar con respecto a los métodos tradicionales.

La diferencia principal entre un enfoque tradicional y AutoML está en el grado de automatización. En un proceso tradicional, cada decisión se realiza de forma manual: elegir el algoritmo, crear transformaciones, ajustar hiperparámetros y comparar modelos. En AutoML, muchas de esas tareas se automatizan para acelerar la búsqueda de una solución inicial.

Esto aporta varias ventajas. La primera es la rapidez. En lugar de probar manualmente cinco o seis modelos, la herramienta puede generar muchas alternativas en menos tiempo. La segunda es la comparación ordenada, porque las plataformas suelen presentar rankings de modelos con métricas, parámetros y resultados. La tercera es la reducción de errores repetitivos, especialmente en tareas como validación, selección de características o ajuste de hiperparámetros.

Aun así, AutoML tiene límites. Puede proponer un modelo con buena métrica, pero no siempre entiende el contexto empresarial. Puede optimizar accuracy cuando en realidad interesa recall. Puede seleccionar variables que generan sesgos. Puede obtener buenos resultados en validación y aun así no ser adecuado para producción.

Una comparación sencilla sería esta:

Aspecto	ML tradicional	AutoML
Selección de modelos	Manual.	Automatizada o semiautomatizada.
Ajuste de hiperparámetros	Manual o con búsquedas configuradas.	Integrado en la plataforma.
Velocidad de prueba	Más lenta.	Más rápida.
Control técnico	Mayor control directo.	Menor control, según la herramienta.
Interpretación	Depende del trabajo realizado.	Requiere revisión posterior.
Uso recomendado	Proyectos que necesitan personalización profunda.	Exploración rápida, comparación inicial y entornos productivos controlados.

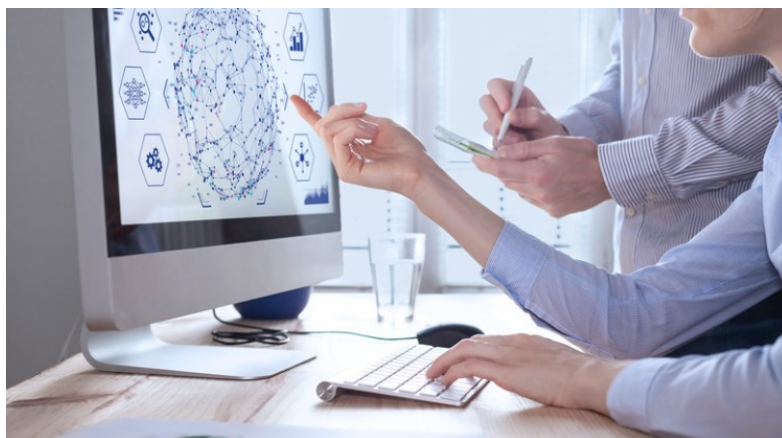
AutoML es especialmente útil cuando se quiere obtener una primera solución, comparar modelos de forma rápida o construir una base sobre la que después se pueda profundizar. Su mejor uso aparece cuando se combina automatización con revisión técnica.

3. Tratamiento de datos desde AWS SageMaker.

AWS SageMaker es una plataforma de Amazon orientada al desarrollo, entrenamiento, despliegue y seguimiento de modelos de Machine Learning. Dentro de este entorno, los datos pueden prepararse, transformarse, analizarse y conectarse con procesos de entrenamiento.

El tratamiento de datos consiste en convertir un conjunto de datos original en una versión útil para el modelo. Esto puede incluir limpieza de valores ausentes, eliminación de duplicados, transformación de columnas, codificación de variables categóricas, normalización, selección de características y separación entre entrenamiento y validación.

En SageMaker, los datos suelen almacenarse en Amazon S3, que actúa como repositorio para archivos de entrada, datasets preparados, artefactos del modelo y resultados del entrenamiento. AWS también ofrece servicios para procesar y transformar datasets antes del entrenamiento; SageMaker Processing permite ejecutar scripts y notebooks para preparar, transformar y analizar datos en un entorno gestionado.



Por ejemplo, una empresa puede subir a S3 un archivo CSV con clientes y comportamiento de uso. Después puede limpiar el archivo, eliminar columnas innecesarias, transformar categorías y preparar el dataset final para AutoML. Esta fase es importante porque AutoML automatiza muchas tareas, pero necesita datos bien estructurados para

generar resultados fiables.

La calidad del dato sigue siendo responsabilidad del proyecto. Si el dataset contiene errores, sesgos o valores incoherentes, la herramienta puede entrenar un modelo técnicamente correcto sobre una base poco fiable.

4. Introducción a la carga y tratamiento de conjuntos de datos sobre el servicio SageMaker.

La carga de datos en SageMaker suele comenzar con la preparación del archivo o fuente de datos. Lo más habitual es trabajar con CSV, Parquet, JSON u otros formatos compatibles con flujos de datos empresariales. El dataset puede proceder de una base SQL, una herramienta de analítica, un CRM, sensores, logs o archivos exportados desde otros sistemas.

Una vez cargados los datos, hay que revisar su estructura. Conviene comprobar columnas, tipos de dato, valores ausentes, distribución de la variable objetivo, duplicados y posibles errores. Por ejemplo, en un modelo de abandono de clientes, habría que revisar si la columna que indica “abandona” está bien definida y si hay suficiente equilibrio entre clientes que abandonan y clientes que permanecen.

En SageMaker, el flujo habitual puede seguir esta lógica:

- Preparar el dataset original.
- Subirlo a S3.
- Revisar columnas y tipos de dato.
- Realizar transformaciones necesarias.
- Seleccionar variable objetivo.
- Configurar el trabajo AutoML.
- Ejecutar entrenamiento.
- Revisar modelos candidatos.
- Validar el modelo seleccionado.

El tratamiento de datos no debe verse como una fase secundaria. En AutoML, una herramienta puede probar muchos modelos, pero si la variable objetivo está mal definida o las columnas contienen información filtrada del futuro, los resultados serán engañosos.

Un ejemplo claro sería entrenar un modelo de abandono usando una columna llamada `fecha_baja`. Si esa columna solo existe cuando el cliente ya se ha dado de baja, el modelo estaría usando una pista que no estará disponible en el momento real de predicción. Esto se conoce como fuga de datos y puede hacer que el rendimiento parezca mucho mejor de lo que será en producción.

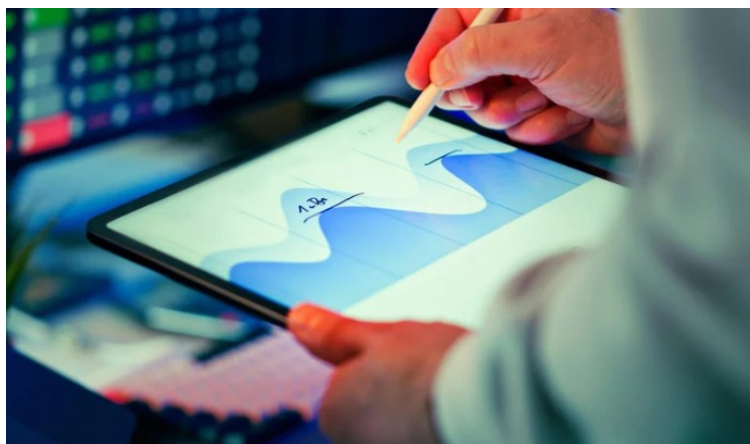
5. Visualización de datos en SageMaker con la finalidad de obtener información sobre el conjunto de datos.

Visualizar datos antes de entrenar un modelo ayuda a entender si el conjunto de datos está preparado para AutoML. Una tabla puede ocultar problemas que un gráfico muestra rápidamente: clases desequilibradas, valores extremos, columnas con demasiados nulos, relaciones sospechosas o distribuciones poco naturales.

En SageMaker, la visualización puede utilizarse para explorar el dataset antes de iniciar el entrenamiento. También puede hacerse mediante notebooks con Python, utilizando pandas, matplotlib, seaborn o herramientas integradas del entorno.

Por ejemplo, antes de crear un modelo de predicción de abandono, conviene visualizar cuántos clientes abandonan frente a cuántos continúan. Si el 98 % continúa y solo el 2 % abandona, el modelo puede obtener una accuracy alta sin detectar bien los abandonos. En ese caso, habría que revisar métricas como recall, precision o F1-score.

También puede visualizarse la distribución de variables numéricas. Si una columna de edad contiene valores como 300 o -5, es probable que haya errores. Si una variable de importe tiene valores extremos, habrá que decidir si son casos reales o registros anómalos. Si una categoría aparece escrita de varias formas, convendrá unificarla.



La visualización también sirve para detectar posibles sesgos. Si un modelo se entrena con datos poco representativos, sus resultados pueden ser peores para ciertos grupos o contextos. Por eso, antes del entrenamiento, conviene revisar si el dataset refleja adecuadamente los casos que aparecerán después en uso real.

6. Entrenamiento de modelos ML mediante AutoML.

El entrenamiento mediante AutoML consiste en dejar que la plataforma explore distintas combinaciones de modelos, transformaciones y configuraciones. En lugar de entrenar manualmente un único algoritmo, AutoML genera varios candidatos y los compara según las métricas definidas.

En SageMaker Autopilot, el flujo AutoML puede automatizar tareas como el análisis del dataset, la selección de algoritmos, la generación de candidatos, el entrenamiento, el ajuste y la comparación de modelos. AWS indica que Autopilot puede entrenar y ajustar modelos, y también utilizar datos transformados en flujos de preparación para entrenar y desplegar modelos.



Esto resulta útil porque permite probar distintas alternativas de forma más rápida. Por ejemplo, para un problema de clasificación, la plataforma puede comparar modelos lineales, árboles, ensamblados u otras técnicas. Para un problema de regresión, puede probar modelos orientados a predecir valores numéricos.

La configuración del entrenamiento debe indicar, como mínimo, el dataset, la

variable objetivo y el tipo de problema si la plataforma no lo detecta automáticamente. Después, la herramienta genera modelos candidatos y muestra resultados comparativos.

6.1. Aplicación práctica y entrenando de modelos sobre SageMaker.

Una aplicación práctica puede partir de un caso sencillo: una empresa quiere predecir qué clientes tienen más riesgo de abandonar un servicio. El dataset contiene columnas como antigüedad, días sin conexión, reclamaciones, compras mensuales, importe medio y una variable objetivo llamada abandono.

El proceso podría plantearse así:

- Primero se prepara el archivo CSV con las columnas necesarias. Después se sube a S3. A continuación, se crea un trabajo AutoML indicando que la variable objetivo es abandono. La plataforma analiza el conjunto de datos, genera candidatos de modelo y ejecuta los entrenamientos.
- Después del entrenamiento, se obtiene una lista de modelos candidatos con sus métricas. En un problema de abandono, no bastaría con elegir el que tenga mayor accuracy. Habría que revisar si detecta correctamente los casos de abandono real. En muchos casos, el recall de la clase positiva puede ser más importante que el acierto global.

Una ventaja de SageMaker es que permite continuar el flujo más allá del entrenamiento. El modelo seleccionado puede registrarse, desplegarse o integrarse con otros procesos. SageMaker Training permite entrenar modelos en infraestructura gestionada y trabajar con algoritmos integrados, scripts propios o frameworks habituales, según la documentación oficial.

6.2. Validación y evaluación de modelos ML sobre SageMaker.

Validar un modelo significa comprobar si funciona bien con datos que no ha utilizado directamente para aprender. Esto evita confiar en un rendimiento inflado por memorización o por fuga de datos.

En AutoML, la plataforma suele separar datos para entrenamiento y validación, o permite configurar esa separación. Aun así, la evaluación debe revisarse con criterio. Las métricas deben estar alineadas con el problema.

En clasificación, pueden revisarse accuracy, precision, recall, F1-score, matriz de confusión y curva ROC. En regresión, se utilizan métricas como MAE, RMSE o R^2 . En problemas desequilibrados, accuracy puede ser insuficiente.



Por ejemplo, en un modelo de abandono, un falso negativo puede ser costoso: el modelo dice que el cliente no abandonará, pero en realidad sí lo hace. Esto impide actuar a tiempo. En un modelo de fraude, dejar pasar una operación fraudulenta puede ser más grave que revisar por error una operación legítima.

La validación también debe incluir revisión de variables utilizadas, estabilidad del modelo, sesgos, comportamiento por segmentos y

posibilidad de mantenimiento. Un modelo puede obtener buena métrica inicial y aun así ser poco adecuado si depende de variables difíciles de actualizar o de datos que no estarán disponibles en producción.

7. Implementación de servicios web ML.

Un modelo entrenado aporta valor cuando puede utilizarse en un proceso real. Para ello, muchas veces se despliega como servicio web. Esto permite que una aplicación envíe datos al modelo y reciba una predicción.

Por ejemplo, una aplicación interna puede enviar los datos de un cliente y recibir como respuesta una probabilidad de abandono. Un sistema de comercio electrónico puede enviar información de un producto y recibir una recomendación. Una herramienta de soporte puede enviar el texto de una incidencia y recibir una categoría.

Un servicio web ML suele funcionar así:

- Una aplicación envía una solicitud con datos de entrada.
- El servicio recibe la solicitud.
- El modelo procesa los datos.
- El sistema devuelve una predicción.

La aplicación utiliza la respuesta para mostrar una alerta, guardar un resultado o activar una acción.

SageMaker permite desplegar modelos para inferencia en endpoints, de forma que puedan recibir solicitudes y devolver predicciones en tiempo real. AWS describe los endpoints de SageMaker como una opción para servir modelos en producción y obtener inferencias desde aplicaciones.

Una API de Machine Learning debe estar bien diseñada. No basta con que devuelva una predicción. Debe validar datos de entrada, controlar errores, registrar solicitudes relevantes, proteger accesos y ofrecer respuestas claras.

Una respuesta de una API podría tener esta estructura:

```
{
  "id_cliente": 125,
  "probabilidad_abandono": 0.82,
  "nivel_riesgo": "alto",
  "modelo": "abandono_clientes_v3"
}
```

Este tipo de respuesta permite que otros sistemas utilicen el resultado del modelo de forma sencilla.

8. Desarrollo de servicios web seguros sobre modelos entrenados en SageMaker accesibles mediante API.

Cuando un modelo se publica mediante API, aparece una nueva responsabilidad: proteger el servicio. Un endpoint de predicción puede recibir datos sensibles, generar decisiones importantes o integrarse con procesos empresariales. Por eso, la seguridad debe considerarse desde el diseño.

Un servicio web seguro debe controlar quién puede acceder, qué datos se envían, cómo se registran las solicitudes y qué ocurre si aparece un error. También debe evitar exponer información sensible en respuestas, logs o mensajes de depuración.

En SageMaker, el despliegue de modelos puede realizarse en endpoints para inferencia en tiempo real. AWS también documenta opciones para desplegar modelos desde el registro mediante el SDK de SageMaker o Boto3, y señala que los proyectos MLOps pueden automatizar despliegues de versiones aprobadas.

Algunas prácticas básicas para servicios web ML seguros son las siguientes:

- Controlar acceso mediante permisos e identidad.
- Cifrar comunicaciones.
- Validar los datos de entrada.
- Evitar enviar datos personales innecesarios.
- Registrar errores sin exponer información sensible.
- Versionar modelos y endpoints.
- Monitorizar uso, latencia y errores.
- Revisar periódicamente calidad de predicciones.
- Preparar un plan de retirada o actualización del modelo.

La seguridad también afecta al modelo. Si cambian los datos de entrada con el tiempo, el modelo puede perder calidad. Si se reciben valores fuera de rango, la predicción puede ser poco fiable. Si el endpoint se usa de forma indebida, puede generar costes o riesgos operativos.

Aquí aparece la relación con MLOps. MLOps busca ordenar el ciclo de vida del modelo: entrenamiento, validación, aprobación, despliegue, monitorización, actualización y retirada. SageMaker incluye capacidades relacionadas con despliegue, seguimiento de linaje, pipelines y monitorización dentro de su ecosistema de Machine Learning.

¡En un flujo profesional, el modelo no debería pasar directamente del experimento a producción! Primero se valida, se registra, se aprueba, se despliega en un entorno controlado, se monitoriza y se actualiza cuando los datos o el rendimiento cambian.



Ejemplo aplicado

Una empresa quiere automatizar la detección de clientes con riesgo de abandono. Primero prepara un dataset con datos históricos: actividad reciente, reclamaciones, facturación, antigüedad y estado final. Después sube el dataset a S3 y utiliza AutoML para generar varios modelos candidatos.

La plataforma entrena modelos, compara métricas y propone una opción con buen rendimiento. El equipo revisa que el modelo no dependa de columnas incorrectas, analiza métricas por segmento y comprueba que detecta bien los abandonos reales.

Después, el modelo se despliega como endpoint. La aplicación interna envía los datos de cada cliente y recibe una probabilidad de abandono. Si el riesgo es alto, se activa una acción de seguimiento.

Finalmente, el modelo se monitoriza. Si con el paso de los meses cambia el comportamiento de los clientes, se revisan los datos y se plantea un nuevo entrenamiento. Así, AutoML no queda como un experimento aislado, sino como parte de un ciclo de mejora y mantenimiento.

AutoML permite automatizar parte del desarrollo de modelos de Machine Learning, acelerar pruebas y comparar alternativas. SageMaker ofrece un entorno para preparar datos, entrenar modelos, evaluar resultados y desplegar soluciones mediante endpoints.

El valor real aparece cuando AutoML se combina con criterio técnico, validación, seguridad y MLOps. Un modelo no debe elegirse solo porque tenga buena métrica, sino porque funciona con datos reales, se puede explicar, se puede mantener y aporta una respuesta útil al problema planteado.

Actividad de relacionar 29

Une cada elemento de la columna A con la opción correcta de la columna B.

Columna A	Columna B
1. AutoML	a. Servicio donde suelen almacenarse datasets, artefactos y resultados en proyectos de AWS.
2. SageMaker Autopilot	b. Prácticas orientadas a gestionar el ciclo de vida de modelos desde entrenamiento hasta despliegue y monitorización.
3. Amazon S3	c. Proceso de comprobar si el modelo funciona bien con datos no utilizados directamente para aprender.
4. Variable objetivo	d. Herramienta de AWS que automatiza parte del desarrollo de modelos de Machine Learning.
5. Validación	e. Conjunto de técnicas que automatizan selección, entrenamiento, ajuste y comparación de modelos.
6. Endpoint	f. Dato que el modelo intenta predecir, como abandono, ventas o riesgo.
7. API	g. Punto de acceso donde un modelo desplegado puede recibir datos y devolver predicciones.
8. MLOps	h. Interfaz que permite a una aplicación comunicarse con un servicio o modelo.
9. Fuga de datos	i. Situación en la que el modelo usa información que no estaría disponible en el momento real de predicción.
10. Monitorización	j. Seguimiento del rendimiento, errores y comportamiento del modelo una vez desplegado.

9. Resumen.

El Módulo 7 ha presentado el Auto Machine Learning como evolución natural del desarrollo de modelos de inteligencia artificial, orientada a automatizar las tareas repetitivas y costosas que tradicionalmente consumían la mayor parte del tiempo del científico de datos. La selección de algoritmos, la optimización de hiperparámetros, la ingeniería de características y la validación de modelos pueden ahora delegarse parcial o totalmente a sistemas automáticos que exploran de forma sistemática el espacio de soluciones posibles.

Se han revisado las principales plataformas de AutoML disponibles en el mercado, tanto las soluciones de los grandes proveedores de nube como Google Cloud AutoML, Azure Machine Learning o Amazon SageMaker Autopilot, como las alternativas de código abierto como Auto-sklearn, H2O AutoML, TPOT o AutoGluon. Cada una de estas herramientas presenta características distintivas en cuanto a los algoritmos que explora, las estrategias de búsqueda que emplea y el grado de control que ofrece al usuario sobre el proceso.

El módulo ha analizado las ventajas del AutoML, entre las que destacan la democratización del acceso al aprendizaje automático para perfiles con menor experiencia técnica, la aceleración de los ciclos de desarrollo, la reducción del riesgo de sesgos derivados de elecciones subjetivas y la posibilidad de obtener líneas base sólidas en muy poco tiempo. Estas ventajas explican la rápida adopción del AutoML en organizaciones de todos los tamaños y sectores de actividad.

También se han discutido las limitaciones y los riesgos asociados al uso indiscriminado del AutoML. La automatización no exime al profesional de comprender los problemas que aborda, de validar críticamente los modelos obtenidos y de asumir la responsabilidad sobre las decisiones que estos toman. El AutoML es una herramienta complementaria al criterio humano, no un sustituto del mismo, y su uso debe enmarcarse en metodologías rigurosas de desarrollo y evaluación de sistemas de inteligencia artificial.

Finalmente, el módulo ha apuntado las tendencias actuales del AutoML, incluyendo su integración con el paradigma MLOps para automatizar no solo el entrenamiento sino también el despliegue, la monitorización y la actualización continua de los modelos en producción. La convergencia entre AutoML, modelos preentrenados y técnicas de ajuste fino sobre arquitecturas como los transformadores está redefiniendo las prácticas del aprendizaje automático en la práctica profesional contemporánea.

10. Actividades tipo test.

1. ¿Qué automatiza fundamentalmente AutoML?

- a) La adquisición de hardware especializado.
- b) El proceso completo de selección, entrenamiento y ajuste de modelos.
- c) La interpretación legal de la normativa de IA.
- d) La elaboración manual de informes ejecutivos.

2. ¿Cuál de las siguientes NO es una plataforma típica de AutoML?

- a) Google Cloud AutoML.
- b) Auto-sklearn.
- c) Apache Hadoop.
- d) H2O AutoML.

3. ¿Qué tarea forma parte de un pipeline de AutoML?

- a) Búsqueda de hiperparámetros.
- b) Recogida de testimonios cualitativos.
- c) Diseño gráfico del producto.
- d) Cálculo del IRPF.

4. ¿Qué ventaja principal aporta AutoML a perfiles no expertos?

- a) Eliminación total de la necesidad de comprender el problema.
- b) Democratización del acceso al aprendizaje automático.
- c) Garantía absoluta de modelos sin sesgos.
- d) Reducción del coste de los datos.

5. ¿Qué limitación importante tiene AutoML?

- a) No puede entrenar modelos de regresión.
- b) No exige de validar críticamente los modelos obtenidos.
- c) Solo funciona con bases de datos relacionales.
- d) Es incompatible con técnicas de deep learning.

6. ¿Qué técnica suele usar AutoML para encontrar buenos hiperparámetros?

- a) Búsqueda bayesiana o aleatoria.
- b) Resolución manual por prueba y error.
- c) Encuestas a los usuarios finales.
- d) Análisis grafológico.

7. ¿Qué se entiende por meta-aprendizaje en AutoML?

- a) Aprender de la experiencia previa con otros problemas para configurar mejor el actual.
- b) Aprender únicamente con datos sintéticos.
- c) Aprender sin función de pérdida.
- d) Aprender de forma adversarial entre dos redes.

8. ¿Con qué paradigma se integra cada vez más AutoML para automatizar despliegue y monitorización?

- a) DevOps clásico exclusivamente.
- b) MLOps.
- c) Lean Startup.
- d) Six Sigma.

9. ¿Cuál es un riesgo del uso indiscriminado de AutoML?

- a) Encarecer drásticamente todos los proyectos.
- b) Tomar decisiones sin comprender los modelos resultantes.
- c) Imposibilitar la reproducibilidad de los experimentos.
- d) Excluir el uso de redes neuronales.

10. ¿Qué tipo de modelos suele incluir AutoML moderno además de los clásicos de ML?

- a) Modelos basados en transformadores y deep learning.
- b) Modelos jurídicos basados en jurisprudencia.
- c) Modelos contables de partida doble.
- d) Modelos hidráulicos analógicos.

Responsible AI

1. Comprender y articular los problemas críticos, sociales, legales, políticos y éticos que surgen a lo largo del ciclo de vida de los datos.

Un proyecto de IA no empieza cuando se entrena un modelo. Empieza mucho antes, cuando se decide qué problema se quiere resolver, qué datos se van a recoger, de dónde proceden, quién aparece representado en ellos y qué decisiones se tomarán después con los resultados.

A lo largo del ciclo de vida de los datos pueden aparecer problemas importantes. Algunos son técnicos, como errores, duplicados o datos incompletos. Otros son sociales y éticos, como sesgos, exclusiones o usos que pueden afectar injustamente a ciertas personas. También existen riesgos legales si se tratan datos personales sin base adecuada, si se recogen más datos de los necesarios o si no se informa correctamente sobre su uso.



El ciclo de vida de los datos puede entenderse como una cadena. Si un eslabón falla, el resultado final se ve afectado.

- Primero se recopilan datos. Aquí puede surgir un problema si los datos proceden de fuentes poco fiables, si se recogen sin consentimiento cuando es necesario o si se incluyen datos sensibles sin una justificación clara.
- Después se almacenan y organizan. En esta fase importan la seguridad, los permisos de acceso, la trazabilidad y la conservación. No

tiene sentido guardar información indefinidamente “por si algún día sirve”. La acumulación innecesaria aumenta riesgos.

- Más tarde se preparan los datos para el modelo. Aquí pueden aparecer decisiones delicadas: eliminar registros, imputar valores ausentes, transformar variables o agrupar categorías. Una limpieza mal planteada puede borrar señales importantes o reforzar sesgos.

Durante el entrenamiento, el modelo aprende patrones. Si los datos históricos contienen desigualdades, el modelo puede reproducirlas. Por ejemplo, si una empresa ha contratado tradicionalmente a perfiles muy similares, un sistema de selección entrenado con esos datos podría valorar peor a candidatos diferentes, aunque sean adecuados.

Finalmente, el modelo se despliega y se usa. En esta fase aparecen preguntas importantes: ¿Quién supervisa las decisiones? ¿Se puede reclamar? ¿Se puede explicar el resultado? ¿Qué ocurre si el modelo falla? ¿Cada cuánto se revisa?

Una IA responsable exige mirar todo el proceso, no solo el resultado final.

2. Comprender conceptos relevantes, que incluyen: ética, moralidad, responsabilidad, derechos digitales, gobernanza de datos, interacción persona-datos, investigación e innovación responsables.

La ética en IA se refiere a la reflexión sobre lo que se debe hacer, no solo sobre lo que se puede hacer. Que una tecnología permita automatizar una decisión no significa que sea adecuado automatizarla sin límites.

La moralidad se relaciona con los valores, normas y criterios que una sociedad, una organización o una profesión considera aceptables. En IA, esto se traduce en preguntas muy concretas: ¿Es justo usar estos datos? ¿Es proporcionado este sistema? ¿Puede perjudicar a alguien? ¿Quién responde si se equivoca?

La responsabilidad implica asumir las consecuencias del diseño y uso del sistema. No basta con decir que “lo decidió el algoritmo”. Detrás de una IA hay personas y organizaciones que han definido objetivos, elegido datos, configurado modelos y permitido su aplicación.

Los derechos digitales hacen referencia a la protección de las personas en entornos tecnológicos. En IA, se relacionan con la privacidad,

la protección de datos, la transparencia, la no discriminación, la seguridad y la posibilidad de comprender o cuestionar ciertas decisiones automatizadas. La Comisión Europea recuerda que el RGPD protege los datos personales con independencia de la tecnología utilizada, tanto en tratamientos automatizados como manuales cuando los datos están organizados según criterios definidos.

La gobernanza de datos es el conjunto de normas, roles y procesos que permiten gestionar los datos de forma ordenada. Incluye saber quién puede acceder, cómo se documentan los datos, qué calidad tienen, cuánto tiempo se conservan, cómo se protegen y con qué finalidad se utilizan.

La interacción persona-datos se refiere a la relación entre las personas y la información que se recoge sobre ellas. No se trata solo de datos “fríos”. Detrás de un registro puede haber hábitos, preferencias, problemas, decisiones, comportamiento laboral, salud, ubicación o situación económica.

La investigación e innovación responsables proponen desarrollar tecnología evaluando sus consecuencias desde el inicio. Esto implica anticipar riesgos, escuchar a partes afectadas, revisar impactos, documentar decisiones y corregir problemas antes de que el sistema cause daño.

