Primera parte

1. **Explica cómo implementarías un modelo de machine learning que debe ser explicable y auditable en un entorno regulado (ej. sector financiero o salud).**

Utilizar modelos interpretables como Regresiones lineales o árboles de decisión, o aplicar técnicas de interpretabilidad a modelos complejos, como LIME o SHAP. Mantener registros detallados de decisiones, datos de entrenamiento y pruebas, y justificar cada paso en el modelo. Documentar el flujo de datos y las transformaciones para asegurar trazabilidad en auditorías.

1. No tantas variables, ser claros con las transformaciones de las variables, modelos sencillos para explicar. No fugar información personal y anonimizar. Mantener los datos originales distinguidos y separados de los datos transformados. Copias de seguridad. No guardar datos en computadores personal. Todo trabajar conexiones seguras a la nube.
2. **¿Cómo diseñarías un experimento de A/B Testing para evaluar el impacto de un nuevo modelo de machine learning en producción?**

Dividir aleatoriamente a los usuarios en dos grupos (A y B). El grupo A recibe el modelo actual y el grupo B el nuevo modelo. Medir métricas clave como precisión, recall, tasa de conversión, etc. Realizar análisis estadístico (ej. t-test) para comparar las métricas entre ambos grupos y asegurar que las diferencias observadas sean significativas. \_Extender el análisis en el tiempo, cómo va evolucionando\_

1. **En un modelo basado en redes neuronales profundas, ¿cómo manejarías el problema de vanishing/exploding gradients y qué soluciones aplicarías?**

Para vanishing gradients, usar activaciones ReLU y técnicas de normalización de lotes (batch normalization). Para exploding gradients, implementar clipping de gradientes y usar inicialización adecuada de pesos (por ejemplo, Xavier o He). Además, emplear redes LSTM o GRU en tareas secuenciales.

1. **¿Cómo diseñarías un sistema de recomendaciones en producción que pueda actualizarse dinámicamente con feedback del usuario?**

Implementar un modelo online learning que se actualice a medida que los usuarios proporcionan feedback (clicks, valoraciones). Usar técnicas como filtrado colaborativo y métodos híbridos, y almacenar el feedback en una base de datos actualizada en tiempo real. Realizar actualizaciones periódicas del modelo en batch o incrementalmente.

Pipeline, que cada cierto tiempo reentrene el modelo y evalúe el desempeño del modelo.

1. **¿Cómo aplicarías modelos de machine learning en un entorno con datos altamente heterogéneos y no estructurados (ej. imágenes, texto y datos tabulares)?**

Preprocesar y normalizar los diferentes tipos de datos (imágenes, texto, tabulares). Usar redes neuronales profundas que integren diferentes entradas (como redes convolucionales para imágenes y LSTM/transformers para texto). Combinar resultados con una capa final de fusión de características y un modelo de XGBoost o redes neuronales para tomar decisiones finales. Complementar Texto: NLP, análisis de sentimiento, de discurso, entidades nombradas.

1. **¿Cuáles son las ventajas y desventajas de desplegar un modelo de machine learning como un microservicio en comparación con integrarlo directamente en una aplicación monolítica?**

Microservicio: Conectar varias cosas, código para leer, uno para entrenar, uno para evaluar.

Monolítico: Todo en un solo “archivo”.

Ventajas de microservicios: Escalabilidad, mantenimiento independiente, despliegue autónomo de cada componente. Desventajas: Complejidad en orquestación, sobrecarga de comunicación entre servicios. Ventajas de monolíticos: Simplicidad, menor latencia de comunicación interna. Desventajas: Dificultad para escalar, menos flexibilidad para desplegar actualizaciones independientes, más sensible antes los fallos (se sabe dónde ocurren fallas).

1. **¿Cómo manejarías la monitorización y logging de un modelo en producción para detectar degradaciones en su rendimiento?**

Utilizar herramientas como Prometheus y Grafana para monitorizar métricas de rendimiento (tiempo de respuesta, tasa de error) y métricas del modelo (precisión, recall). Implementar logging detallado con ELK Stack para almacenar eventos y excepciones. Usar alertas automáticas si el rendimiento cae bajo un umbral definido. MLflow\_ herramienta para programar corridas en el tiempo, y envío de alertas a partir de los resultados.

1. **Explica cómo diseñarías una arquitectura de CI/CD para modelos de machine learning en producción.**

Usar herramientas como GitLab CI, Jenkins o Kubeflow para automatizar el pipeline de desarrollo y despliegue. Incluir pasos de validación de modelos (tests unitarios, integración, evaluación de rendimiento), entrenamiento en un entorno controlado, y despliegue automatizado en producción. Versionar los modelos y realizar pruebas de regresión. Buscar mierda.

1. **¿Cómo optimizarías el tiempo de inferencia de un modelo en producción sin comprometer su precisión?**

Desde el back-end optimizar el código para la ingesta de datos, haciéndolos más ligeros. Tener una infraestructura robusta que responda a las peticiones que se esperan recibir. Usar las soluciones de proveedores cloud, que están diseñadas para optimizar el tiempo de respuesta.

Aplicar pruning (recorte) de redes neuronales para reducir el tamaño del modelo, quantization para representar los pesos con menor precisión, y usar técnicas como TensorRT o ONNX para optimizar el modelo para la inferencia en producción. También se pueden usar modelos distilados o versiones más pequeñas del modelo original.

1. **Explica cómo manejarías la orquestación de workflows de ML en la nube y qué herramientas utilizarías.**

Utilizar herramientas como Kubeflow, Airflow, o MLflow para orquestar los pipelines de ML en la nube. Estas plataformas permiten integrar tareas de preprocesamiento, entrenamiento, validación y despliegue en un solo flujo de trabajo automatizado, gestionando dependencias, y realizando monitorización en tiempo real. Utilizar AWS SageMaker o Google AI Platform para gestionar el ciclo de vida de modelos en la nube.