Google

# Directrices para trabajar con aprendizaje automático de manera responsable

## Directrices

### Toma precauciones cuando los flujos de control se modifiquen debido a los resultados de los modelos de aprendizaje automático.

Los flujos de trabajo que consumen la salida de una característica básica del aprendizaje automático pueden ser modificados para generar superficies de ataque que sean difíciles de asegurar. Tal como se aconseja actualmente, no se recomienda incorporar datos en un flujo de trabajo de ejecución si es que no se pide explícitamente.

### Evita el acceso directo a los modelos y los metadatos del modelo.

La interacción directa con las características básicas del aprendizaje automático puede dar lugar a una cantidad de superficies de ataque diferentes, como la exfiltración, la inversión de datos y muchas otras. Mantén todo esto fuera del alcance de los usuarios y solamente permite las interacciones controladas y aprobadas, y garantiza que la salida esté conectada con una característica establecida, como una base de datos. No le brindes datos de salida al usuario de manera directa.

### Garantiza que los modelos utilicen controles de integridad tanto en el entrenamiento como luego de la implementación.

Se pueden producir modificaciones en el software durante las distintas etapas del proceso de creación y despliegue; lo mismo ocurre con el aprendizaje automático. Revisa y luego revisa otra vez. Solo ejecuta lo que sabes que se espera. Autentifica antes de operar.

### Si las características son complejas, es posible que los controles también lo sean.

Las interacciones deben ser breves, directas y concisas. Los flujos de trabajo extensos presentan más riesgos de exposición en ataques exitosos.

### Comienza con una lógica extensional para el entrenamiento y, luego, cuando sea necesario, continúa con una lógica intensional para las características y la inferencia.

Utilizar un conjunto validado y conocido en el entrenamiento es un buen modo de evitar la contaminación de datos y otros ataques. Sin embargo, para crear características con mayores capacidades de realizar inferencias, se deben incorporar nuevos datos; probar minuciosamente los resultados de este entrenamiento basándose en los resultados esperados de las entradas válidas y conocidas genera sistemas más resilientes.

### Asegura que las pruebas de integración se creen junto con el caso práctico previsto de la característica básica.

Los flujos de trabajo del aprendizaje automático requieren pruebas de integración y de unidades, como cualquier otro software. La naturaleza heurística del software de aprendizaje automático puede hacer que la construcción de afirmaciones confiables sea difícil, por lo que recomendamos que te centres en los resultados del flujo de trabajo deseados a largo plazo.

### Busca ataques que utilicen síntesis automática, en particular, en los controles que se centran en la identidad.

Siempre pregúntate lo siguiente: "¿una máquina podría hacer esto, ahora?" antes de suponer que una voz, un video o un texto son auténticos, especialmente cuando se utilicen para otorgar derechos de acceso.

### Utiliza políticas mutuas para mantenerte al tanto del modelo económico y así garantizar un modelo de seguridad sólido y de bajo costo.

Si el costo de un ataque es menor que el costo de su reparación posterior, entonces es probable que el control sea ineficaz. Implementa que los costos de seguridad se ajusten automáticamente según el impacto de los ataques exitosos para aumentar de modo gradual los costos del atacante.

Google

### Evita que el usuario controle los modelos entrenados con datos propietarios, como en un dispositivo.

Considera el caso de uso en la creación de la característica básica del aprendizaje automático e imagina que, en teoría, los datos de entrenamiento que la crearon se pueden recuperar. Utiliza esto cuando planifiques la continuidad del negocio y en las evaluaciones de riesgos cuando desarrolles un producto.

### Utiliza un límite de tasas para evitar ataques de extrapolación y el consumo innecesario de recursos.

Los controles tradicionales siguen siendo muy efectivos; por ejemplo, los controles en capas, como las listas de control de acceso, el límite de tasas y la autenticación al diseñar e implementar tecnologías de aprendizaje automático.