

FINE-TUNING CON OPENAI



De datos a modelo



DATASPHERE
www.datasphere.tech

Nelson Zepeda
nelson.zepeda@datasphere.tech
Junio 2025

ACERCA DE DATASPHERE

Datasphere es un proveedor pionero en soluciones avanzadas de análisis y IA, diseñadas para transformar organizaciones en empresas impulsadas por datos.

Fundada en 2019, está posicionada para aprovechar la creciente demanda de soluciones basadas en datos.

Utilizando herramientas analíticas modernas y enfoques centrados en el cliente, ayudamos a transformar modelos de negocio y lograr objetivos estratégicos.



Datasphere ha desarrollado con éxito proyectos en El Salvador, Honduras, Bolivia, Tel Aviv, EE. UU. y Sudáfrica.



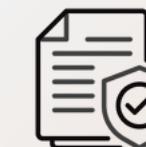
BIG DATA



**BUSINESS
INTELLIGENCE**



**AI / MACHINE
LEARNING**



**DATA
GOVERNANCE**



**STAFF
AUGMENTATION**



TRAININGS

NUESTRO STACK



ANTHROPIC





AGENDA



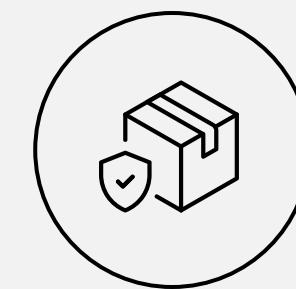
Conceptos



Fine-Tuning vs ML

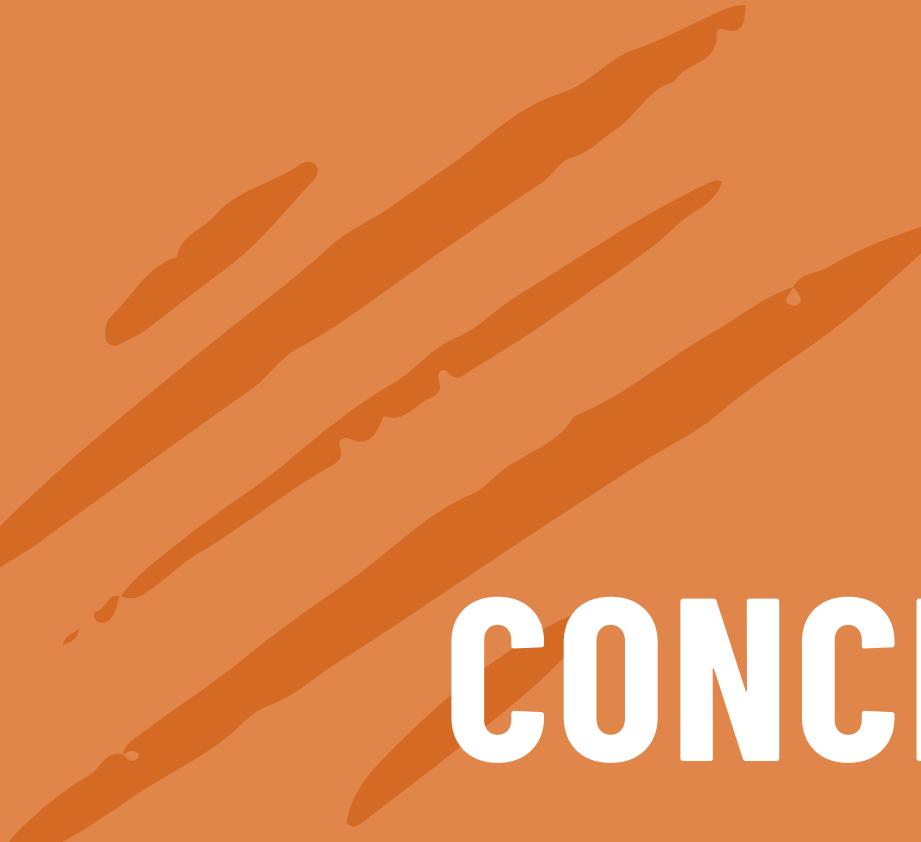


Proceso en OpenAI



Hiperparametros





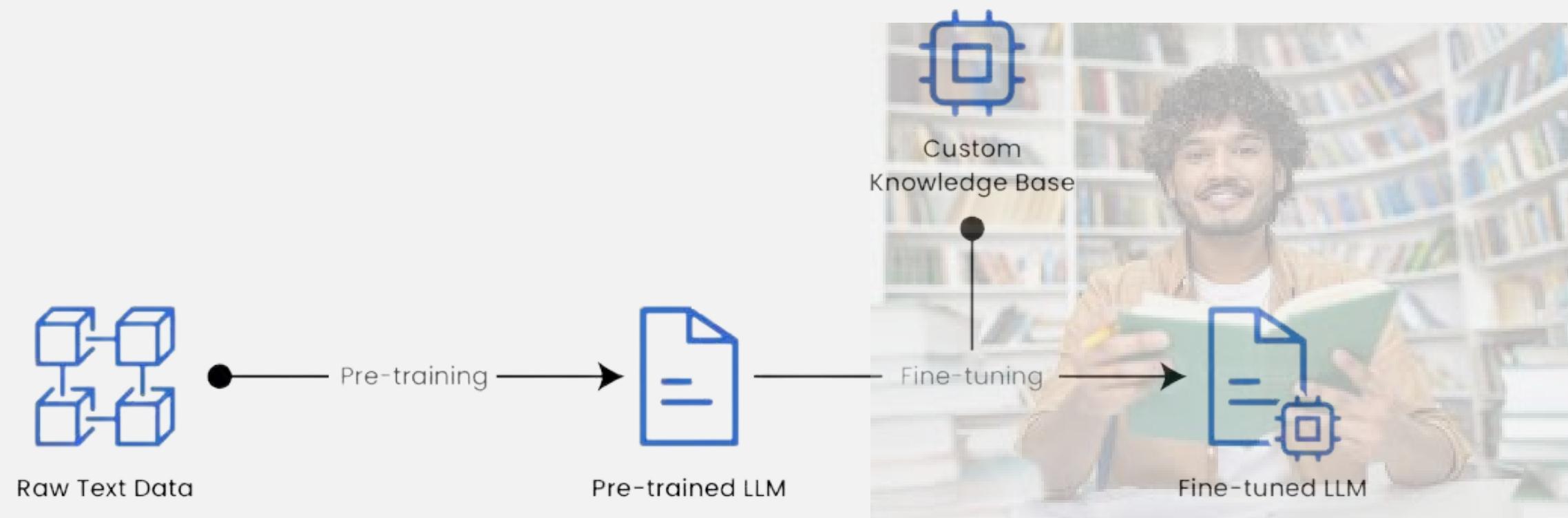
CONCEPTOS

FINE-TUNING

El fine-tuning es un proceso en el aprendizaje automático donde tomas un modelo pre-entrenado (que ya ha sido entrenado en una cantidad masiva de datos y ha aprendido patrones generales del lenguaje) y lo adaptas a una tarea o conjunto de datos más específico.

Imagina que tienes un estudiante muy inteligente (el modelo pre-entrenado) que ha leído una biblioteca entera de libros sobre todos los temas del mundo. Este estudiante es muy bueno en comprensión general, redacción y lógica.

Sin embargo, si lo quieres convertir en un experto en "historia del arte renacentista", no vas a empezar desde cero enseñándole a leer y escribir. En cambio, le darías solo libros sobre el Renacimiento y le pedirías que se enfoque específicamente en ese tema. Eso es el fine-tuning.

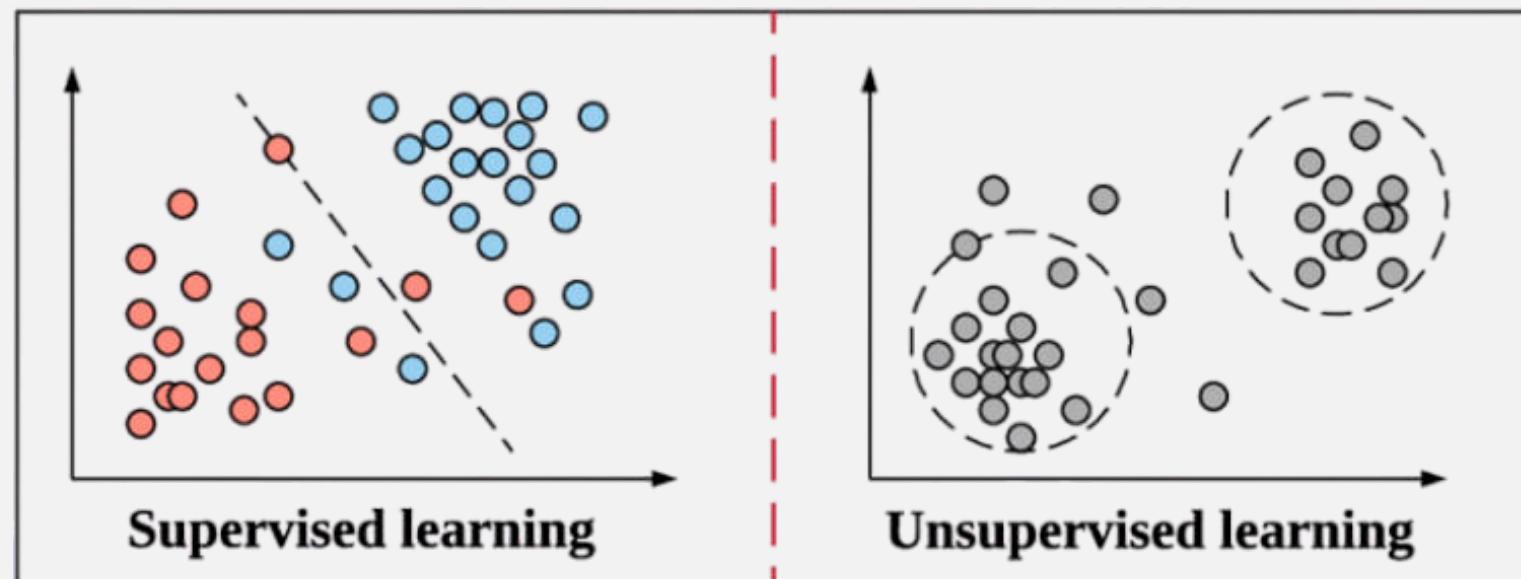


FINE-TUNING

Método	Cómo Funciona	Casos de Uso
Fine-tuning Supervisado (SFT)	Proporciona ejemplos de respuestas correctas a las instrucciones (prompts) para guiar el comportamiento del modelo. A menudo utiliza respuestas "ground truth" (generadas por humanos y consideradas correctas) para mostrar al modelo cómo debe responder.	Clasificación Traducción con matices (matizada) Generación de contenido en un formato específico Corrección de fallos en el seguimiento de instrucciones para prompts complejos (instrucciones complejas).
Optimización Directa de Preferencias (DPO)	Para un prompt (instrucción), proporciona tanto un ejemplo de respuesta correcta como un ejemplo de respuesta incorrecta. Indicar la respuesta correcta ayuda al modelo a rendir mejor cuando la salida correcta es más subjetiva.	Resumir texto, enfocándose en los aspectos correctos (los que importan). Generar mensajes de chat con el tono y estilo adecuados.
Fine-tuning por Refuerzo (RFT)	Solo para modelos de razonamiento: Genera una respuesta para un prompt (instrucción), proporciona una calificación de experto para el resultado, y utiliza la puntuación resultante para reforzar la "cadena de pensamiento" del modelo para respuestas con puntuaciones más altas. Funciona cuando los expertos calificadores pueden ponerse de acuerdo sobre la salida ideal del modelo.	Tareas complejas específicas de un dominio que requieren razonamiento avanzado. Diagnóstico médico basado en el historial y las guías de diagnóstico. Determinar pasajes relevantes de jurisprudencia (leyes y casos legales).

SUPERVISED FINE-TUNING

El SFT te permite enseñar a un modelo de OpenAI a manejar mejor tus casos de uso específicos, entrenándolo con ejemplos que tú proporcionas. El resultado es un modelo personalizado que produce de manera más fiable el estilo y el contenido que deseas.



JSON Lines es un formato de archivo práctico y eficiente para almacenar datos estructurados registro por registro.

Características clave:

- Procesamiento por línea: Cada línea es un registro JSON completo y válido, lo que facilita su procesamiento individualmente.
- Compatibilidad: Funciona muy bien con herramientas de texto de estilo Unix y pipelines de shell, además de ser útil para el intercambio de mensajes entre procesos.
- Archivos de registro: Es un formato excelente y flexible para archivos de registro (logs).

Requisitos esenciales:

1. Codificación UTF-8: Asegura la correcta lectura de caracteres, evitando problemas de interpretación. No debe incluir la Marca de Orden de Bytes (BOM).
2. Cada línea es un valor JSON válido: Lo más común son objetos o arrays, pero cualquier valor JSON es aceptado (ej. null es válido, una línea en blanco no).
3. Separador de línea es '\n': Esto permite que se procesen los archivos línea por línea.

VISION FINE-TUNING

El **Vision fine-tuning** es una técnica que permite ajustar los modelos de OpenAI para que logren una mejor comprensión de las entradas de imagen. En esencia, toma un modelo que ya puede ver e interpretar imágenes (un modelo multimodal) y lo especializa para tareas o tipos de imágenes muy concretos.

Las imágenes deben estar en un formato específico (por ejemplo, rutas a URLs o imágenes base64 codificadas dentro del JSONL, o algún otro formato que OpenAI defina para este tipo de fine-tuning

```
{  
  "messages": [  
    {  
      "role": "system",  
      "content": "Eres un asistente que identifica quesos poco comunes."  
    },  
    {  
      "role": "user",  
      "content": "¿Qué es este queso?"  
    },  
    {  
      "role": "user",  
      "content": [  
        {  
          "type": "image_url",  
          "image_url": {  
            "url": "https://upload.wikimedia.org/wikipedia/commons/3/36/Danbo_Cheese.jpg"  
          }  
        }  
      ]  
    },  
    {  
      "role": "assistant",  
      "content": "Danbo"  
    }  
  ]  
}
```

DPO FINE-TUNING

El ajuste fino por Optimización Directa de Preferencias (DPO) te permite afinar modelos basándose en instrucciones (prompts) y pares de respuestas.

Cómo funciona este enfoque: Este enfoque permite que el modelo aprenda a partir de preferencias humanas más subjetivas, optimizando así las salidas para que sean las más probables de ser favorecidas.

Limitación Actual: Actualmente, DPO solo es compatible con entradas y salidas de texto.

Formato de Datos: Cada ejemplo en tu conjunto de datos (dataset) para DPO debe contener:

- Una instrucción (Prompt): Por ejemplo, un mensaje de usuario.
- Una salida preferida: Que es una respuesta ideal del asistente.
- Una salida no preferida: Que es una respuesta subóptima (menos ideal) del asistente.

```
{  
  "input": {  
    "messages": [  
      {  
        "role": "user",  
        "content": "Hola, ¿me puedes decir qué tan frío hace en San Francisco hoy?"  
      }  
    ],  
    "tools": [],  
    "parallel_tool_calls": true  
  },  
  "preferred_output": [  
    {  
      "role": "assistant",  
      "content": "Hoy en San Francisco, no hace tanto frío como se esperaba. Las nubes matutinas darán paso al sol, con una máxima cercana a los 68°F (20°C) y una mínima alrededor de los 57°F (14°C)."  
    }  
  ],  
  "non_preferred_output": [  
    {  
      "role": "assistant",  
      "content": "No hace mucho frío en San Francisco hoy."  
    }  
  ]  
}
```

REINFORCEMENT FINE-TUNING

El RFT adapta un modelo de razonamiento de OpenAI con una señal de retroalimentación que tú defines.

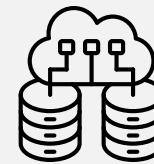
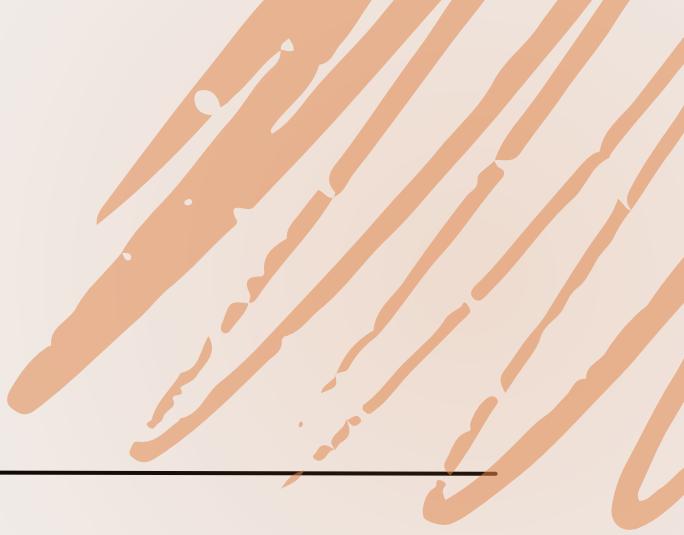
Al igual que el ajuste fino supervisado, personaliza el modelo para tu tarea. La diferencia es que, en lugar de entrenar con respuestas "correctas" fijas, se basa en un calificador programable que puntúa cada respuesta candidata.

El algoritmo de entrenamiento luego ajusta los pesos del modelo, de modo que las salidas con puntuaciones altas se vuelven más probables y las de puntuaciones bajas se desvanecen.

Para realizar el RFT, se debe definir un calificador que puntúa la salida del modelo durante el entrenamiento, indicando la calidad de su respuesta.

```
{  
    "type": "multi",  
    "graders": {  
        "explanation": {  
            "name": "Calificador de texto de explicación",  
            "type": "score_model",  
            "input": [  
                {  
                    "role": "user",  
                    "type": "message",  
                    "content": "...ver otra pestaña para el prompt completo..."  
                }  
            ],  
            "model": "gpt-4o-2024-08-06"  
        },  
        "compliant": {  
            "name": "conforme",  
            "type": "string_check",  
            "reference": "{{item.compliant}}",  
            "operation": "eq",  
            "input": "{{sample.output_json.compliant}}"  
        }  
    },  
    "calculate_output": "0.5 * compliant + 0.5 * explanation"  
}
```

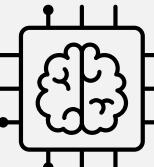
COMPONENTES CLAVE



Modelo Pre-entrenado (Foundation Model): Son modelos gigantes (como GPT-3.5, GPT-4) que han sido entrenados en cantidades masivas de datos de texto y código de internet.



Dataset Específico: Este es tu propio conjunto de datos, mucho más pequeño y especializado, que contiene ejemplos de la tarea específica que quieras que el modelo realice.



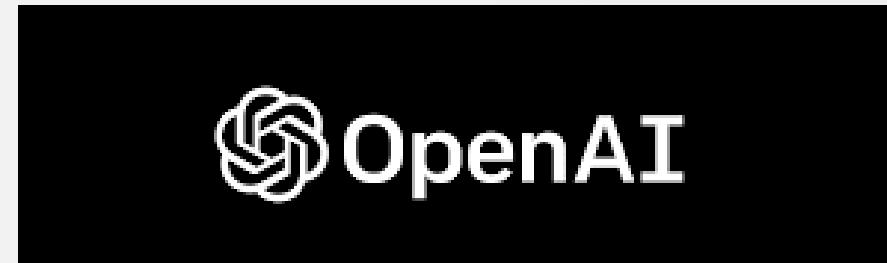
Proceso de Ajuste (entrenamiento): Durante el fine-tuning, el modelo pre-entrenado se entrena adicionalmente con tu dataset específico.



Conexiones: En lugar de "aprender desde cero", el modelo solo "ajusta" sus pesos internos (sus "conexiones neuronales") para optimizar su rendimiento en tu tarea. Es como si el estudiante de nuestra analogía ahora solo viera y memorizara detalles específicos del arte renacentista.

EN LUGAR DE "APRENDER DESDE CERO", EL MODELO SOLO "AJUSTA" SUS PESOS INTERNOS (SUS "CONEXIONES NEURONALES") PARA OPTIMIZAR SU RENDIMIENTO EN TU TAREA. ES COMO SI EL ESTUDIANTE DE NUESTRA ANALOGÍA AHORA SOLO VIERA Y MEMORIZARA DETALLES ESPECÍFICOS DEL ARTE RENACENTISTA.

FINE-TUNING DE LLMS



El ecosistema de modelos de lenguaje grandes (LLMs) está creciendo muy rápido, y cada vez más proveedores y frameworks ofrecen capacidades de fine-tuning.

- Facilidad de uso y modelos de punta: OpenAI, Google Vertex AI, Cohere.
- Flexibilidad, modelos de código abierto y control: Hugging Face.
- Comunidad y experimentación: Hugging Face.
- Infraestructura gestionada para modelos de código abierto: Replicate.
- Control total y privacidad (pero más esfuerzo): Fine-tuning local o en tu propia nube con modelos de código abierto.

Hugging Face

Replicate

ANTHROPIC



FINE-TUNING VS ML

FINE-TUNING VS MACHINE LEARNING

Característica Clave	Machine Learning Supervisado (General)	Fine-tuning de LLMs (Modelos de Lenguaje Grandes)
Punto de Partida del Modelo	Modelo "desde cero" (parámetros aleatorios o con inicialización mínima)	Modelo "pre-entrenado" en vastos datasets de lenguaje (ej. GPT-3.5)
Tipo de Datos Típico	Tabulares (numéricos, categóricos estructurados), imágenes, audio, series de tiempo.	Texto libre , secuencias de lenguaje, conversaciones.
Cantidad de Datos Etiquetados Necesarios	Significativa para que el modelo aprenda desde cero.	Menor cantidad de datos específicos para la tarea, ya que el modelo ya tiene una base de conocimiento.
Costo Computacional (Entrenamiento)	Alto para modelos complejos desde cero.	Generalmente menor que entrenar un modelo similar desde cero, ya que solo se ajustan los pesos.
Tiempo de Entrenamiento	Puede ser largo para modelos complejos y grandes datasets.	Generalmente más rápido porque el modelo ya está pre-entrenado.
Dominio del Conocimiento	Aprende el dominio directamente de tus datos de entrenamiento.	Aprovecha el conocimiento general de lenguaje y el mundo aprendido en el pre-entrenamiento.
Ejemplo de Aplicación Típica	<ul style="list-style-type: none">- Predicción de Churn- Clasificación de clientes- Predicción de precios (regresión)- Detección de fraude (datos transaccionales)- Visión por computadora (clasificación de imágenes)	<ul style="list-style-type: none">- Clasificación de sentimiento- Generación de texto personalizado- Resumen de documentos- Traducción- Chatbots específicos de dominio- Extracción de información de texto
Flexibilidad/Versatilidad	Muy versátil para diferentes tipos de datos estructurados y no estructurados.	Especializado en tareas de lenguaje y texto. Menos adecuado para datos puramente numéricos/tabulares.
Resultado Principal	Un modelo optimizado para tu tarea específica con los datos proporcionados.	Un modelo de lenguaje pre-existente adaptado para un comportamiento, tono o tarea textual específica.
Ejemplo de Herramientas/Algoritmos	Regresión Logística, Árboles de Decisión, Random Forest, XGBoost, LightGBM, scikit-learn, TensorFlow, PyTorch.	OpenAI API (fine-tuning), Hugging Face Transformers, Google Vertex AI, Cohere.

¿QUÉ ES EL CHURN Y CÓMO SE PREDICE TÍPICAMENTE?

los datos son típicamente numéricos y categóricos estructurados en filas y columnas, no texto libre.

Ejemplo de una fila de datos de churn:

Cliente_ID: 123, Edad: 35, Uso_Mensual: 150GB, Quejas: 2, Antigüedad: 24 meses, Churn: 1

Un LLM no está optimizado ni es eficiente para "leer" esta tabla y "entender" las relaciones numéricas o categóricas directamente.

No está diseñado para:

- Interpretar valores numéricos como "150GB" o "24 meses" en un contexto de predicción numérica.
- Calcular correlaciones entre Edad y Uso_Mensual.
- Manejar eficientemente la estructura de datos tabulares.

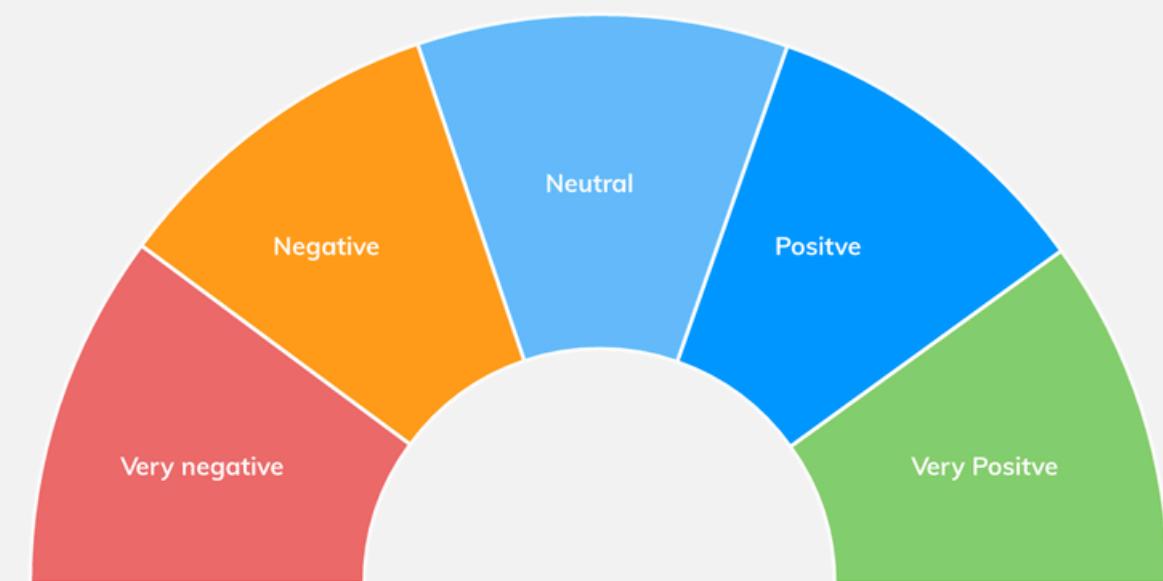
Si intentaras forzar un LLM a hacer esto, tendrías que convertir cada fila de tus datos tabulares en una frase (como "El cliente X tiene 35 años, usa 150GB al mes, tiene 2 quejas y 24 meses de antigüedad. ¿Hará churn? Sí/No").



¿CUÁNDO PODRÍA UN LLM (O FINE-TUNING) SER RELEVANTE PARA EL CHURN?

Los LLMs podrían jugar un papel en un escenario híbrido si tienes datos textuales relacionados con el churn:

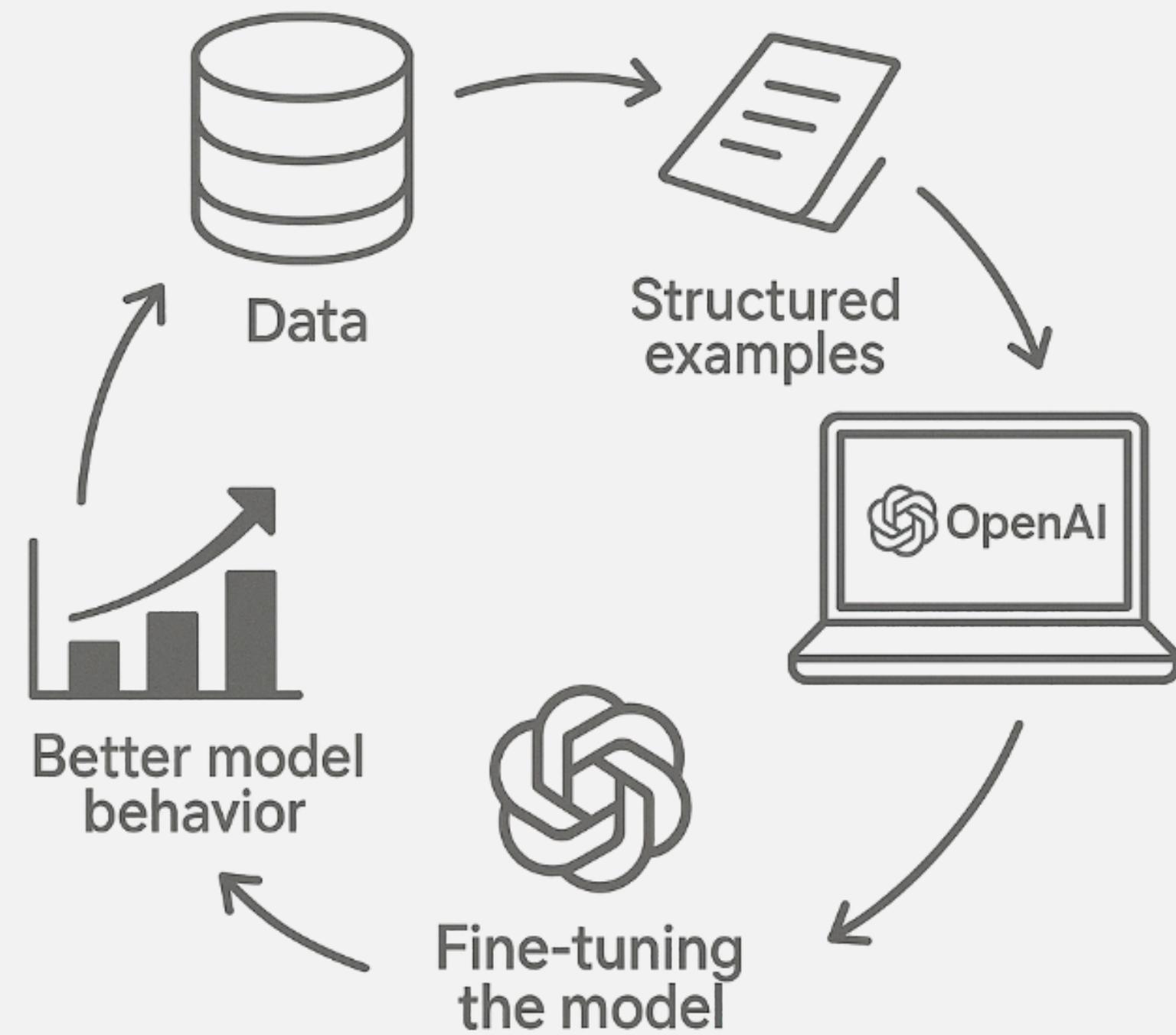
- Clasificación de interacciones con el cliente: Si tienes transcripciones de chats de soporte o grabaciones de llamadas de clientes, un LLM (afinador o no) podría:
 - Clasificar el sentimiento
 - Extraer temas de las quejas
 - Detectar palabras clave o frases que indiquen insatisfacción o intención de cancelar.
 - Resumir el historial de interacciones de un cliente en un texto corto.
 - Generación de texto para la retención: Un LLM podría generar correos electrónicos personalizados para clientes en riesgo de churn, basándose en el historial del cliente.





PROCESO DE FINE-TUNING

FINE-TUNING PROCESO



FINE-TUNING PROCESO



Pasos para Hacer Fine-tuning en la Interfaz Web de OpenAI

 **1. Accede a la Plataforma de OpenAI:**

- Abre tu navegador y ve a <https://platform.openai.com/>.
- Inicia sesión con tu cuenta de OpenAI.

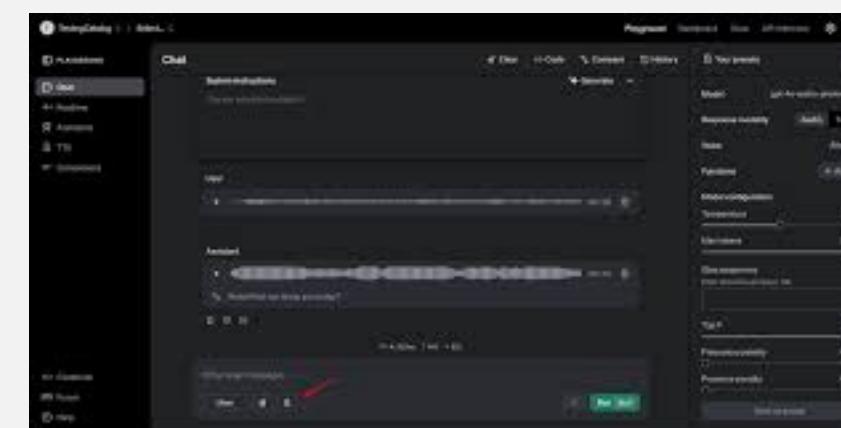
 **2. Navega a la Sección de Fine-tuning:**

- Una vez que hayas iniciado sesión, busca en el menú de la izquierda.
- Deberías ver una opción llamada "Fine-tuning" (o "Ajuste Fino"). Haz clic ahí.

 **3. Cargar tu Archivo de Datos (Upload File):**

- En la página de Fine-tuning, verás un botón que dice "Create" o "Create a fine-tuning job". Haz clic en él.
- Esto te llevará a una pantalla donde puedes configurar el trabajo.
- Te pedirá que elijas un archivo de entrenamiento. Haz clic en "Upload file" o en el área designada para subir archivos.
- Navega en tu computadora y selecciona tu archivo banking_sentiment_finetune_final.jsonl (o el nombre que le hayas dado).
- Despues de seleccionarlo, OpenAI lo subirá y lo procesará. Puede tardar un momento si el archivo es grande. Una vez subido, lo verás listado y te dará un file_id interno.

{JSONL}

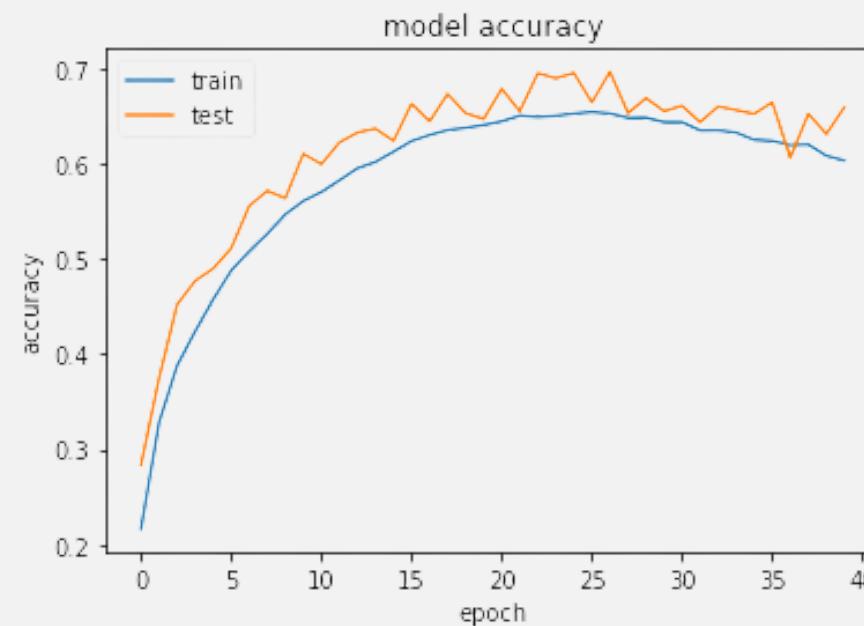


FINE-TUNING PROCESO



4. Configurar y Crear el Trabajo de Fine-tuning:

- Una vez que tu archivo de entrenamiento haya sido subido y validado (verás un mensaje de "ready"), continuarás con la configuración del trabajo:
- "Base model": Elige el modelo base que quieras afinar. Las opciones más comunes son gpt-3.5-turbo o gpt-4o-mini. Selecciona el mismo modelo que utilizaste como referencia en tu script de evaluación si lo sabes.
- "Validation file (optional)": Si tienes un archivo de validación (validation_data.jsonl), puedes subirlo aquí también. Es muy recomendable para monitorear el sobreajuste.
- "Hyperparameters (optional)": Puedes dejar esto en los valores predeterminados ("auto") si no estás seguro. Si eres un usuario avanzado, puedes ajustar n_epochs, batch_size, y learning_rate_multiplier aquí.
- "Suffix (optional)": Puedes añadir un sufijo personalizado al nombre de tu modelo afinado. Esto es útil para identificarlo más fácilmente (ej. my-sentiment-classifier).
- Una vez que hayas configurado todo, haz clic en el botón "Create" o "Create fine-tuning job" en la parte inferior.



FINE-TUNING PROCESO



5. Monitorear el Progreso:

- Despues de crear el trabajo, serás redirigido a una página donde verás el estado de tu fine-tuning.
- El estado comenzará como "pending" o "running".
- Puedes refrescar la página periódicamente o simplemente esperar. OpenAI te enviará un correo electrónico cuando el trabajo se haya completado o si hay algún error.
- En esta página, también podrás ver los logs y las métricas de pérdida (train loss y validation loss si proporcionaste un archivo de validación) a medida que el entrenamiento progresá. Esto te ayuda a entender si el modelo está aprendiendo correctamente.



6. Obtener el ID del Modelo Afinado:

- Cuando el estado del trabajo cambie a "succeeded" (éxito), en la misma página de detalles del trabajo, verás un campo llamado "Fine-tuned model" o similar.
- Ahí estará el ID de tu nuevo modelo afinado (por ejemplo, ft:gpt-3.5-turbo-0125:tu-organizacion::sufijo-personalizado)





HIPERPARÁMETROS

HIPERPARÁMETROS

1

n_epochs

¿Cuántas vueltas completas le da el modelo a tus datos?

2

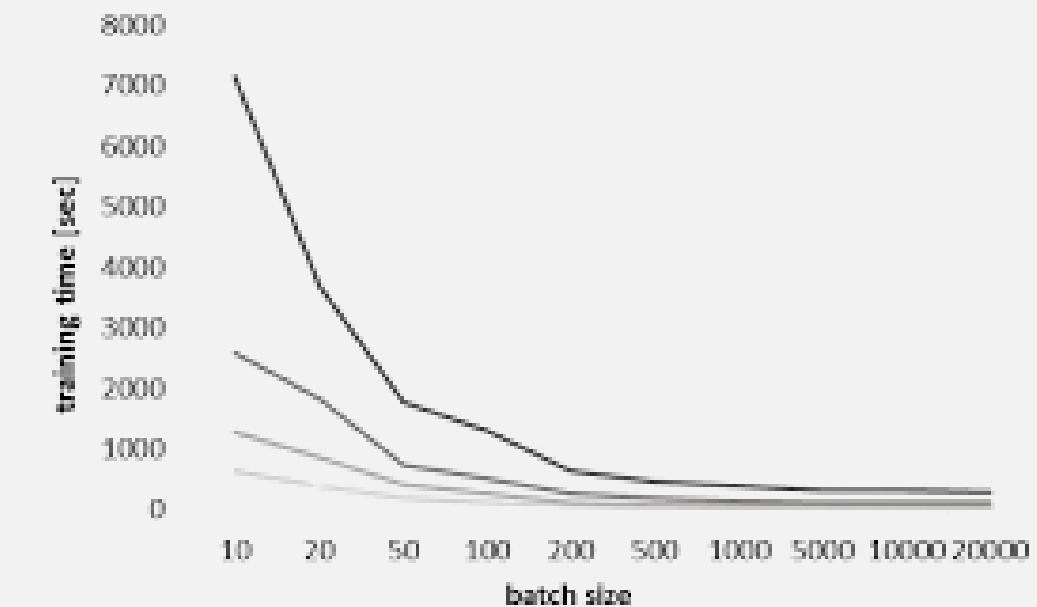
batch_size

¿En qué tamaño de trozos lee el modelo tus datos cada vez?

3

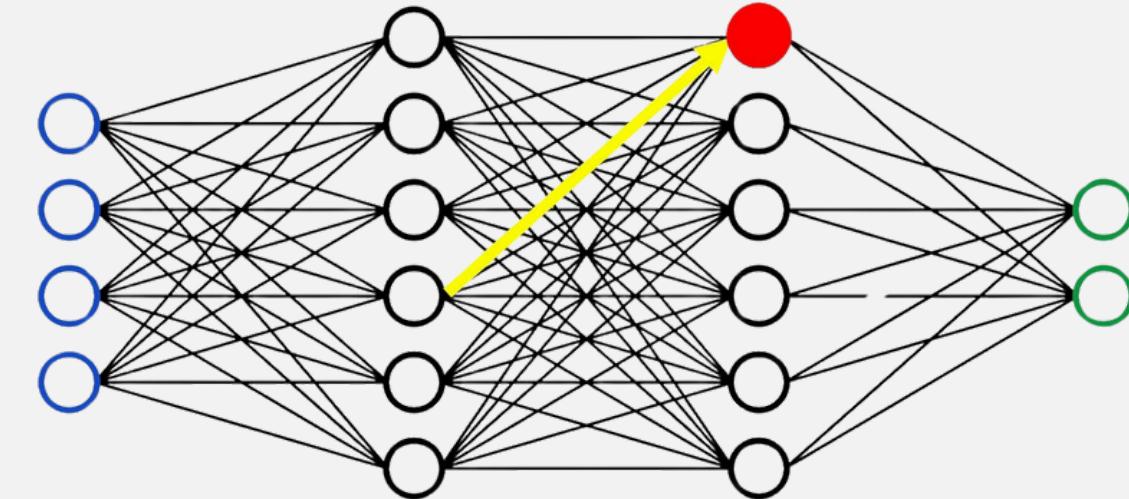
learning_rate_multiplier

¿Qué tan rápido ajusta el modelo lo que aprende de cada trozo? (Velocidad de absorción del conocimiento)

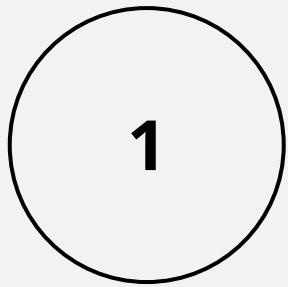


Number of epoch

- 1
- 2
- 3
- 4



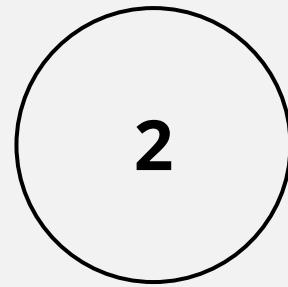
PLAYGROUND



Temperature

Temperatura baja (cerca de 0.0): El modelo elige la palabra siguiente que sea más probable y segura. Las respuestas serán muy deterministas, predecibles y repetitivas.

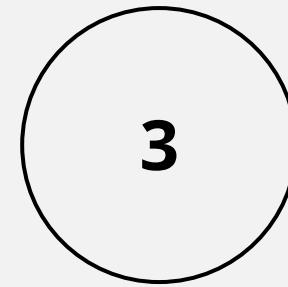
Ideal para tareas de clasificación, extracción de información o cualquier cosa donde quieras respuestas exactas y sin "sorpresa".



Top P

El modelo considera solo las palabras cuya probabilidad sumada es igual o mayor a Top P.

- Top P = 1.0: Considera todas las palabras posibles (más variedad).
- Top P = 0.1: Considera solo las palabras más probables que acumulen el 10% de la probabilidad total (menos variedad, más conservador).



Stop Sequences

Son una lista de cadenas de texto que, si el modelo las genera, detendrán la generación de texto inmediatamente.

Muy útil para controlar el formato de la salida. Por ejemplo, si quieres que cada respuesta sea solo una línea, puedes poner \n como secuencia de parada.

Mode: Chat

Model: gpt-3.5-turbo

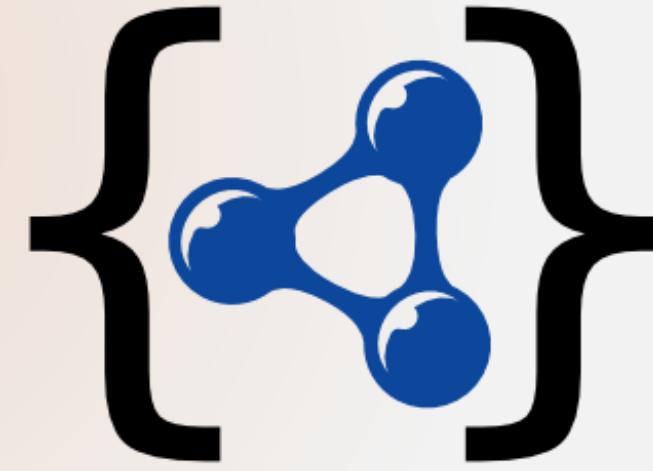
Temperature: 1

Maximum length: 256

Stop sequences: Enter sequence and press Tab

Top P: 1





JSONL

```
{  
  "messages": [  
    {"role": "system", "content": "Tu instrucción general para el modelo."},  
    {"role": "user", "content": "El texto del review del usuario."},  
    {"role": "assistant", "content": "La etiqueta de sentimiento esperada (ej. NEGATIVE, POSITIVE)."}  
]
```



CONCLUSIONES

- **Especialización para Mayor Precisión y Eficiencia:** El Fine-tuning transforma modelos de lenguaje generales (como GPT-3.5) en especialistas altamente eficientes para tareas específicas (ej. clasificación de sentimiento bancario), mejorando drásticamente la precisión y permitiendo prompts más cortos que reducen costos y latencia.
- **Fundamentalmente, es Machine Learning Supervisado:** Aunque avanzado, el Fine-tuning es una forma de ML Supervisado que aprovecha el conocimiento pre-existente del modelo. Se diferencia de entrenar desde cero en que requiere significativamente menos datos etiquetados específicos para ajustar el modelo a tu dominio.
- **No es para Todos los Datos (o Problemas):** El Fine-tuning de LLMs brilla con datos textuales y problemas de lenguaje natural. Para datos tabulares estructurados (como la predicción de churn), los algoritmos de Machine Learning supervisado tradicionales (ej. XGBoost) son mucho más adecuados y eficientes.



MUCHAS GRACIAS



www.datasphere.tech



nelson.zepeda@datasphere.tech



+ 503 75664240



Tribeca Urban Living, Colonia
San Francisco, San Salvador