

Evaluación Técnica – Ingeniero(a) de Machine Learning Junior

Objetivo

Demostrar tu capacidad para construir y exponer modelos de clasificación de texto en un entorno controlado.

Datos

Recibirás un archivo `data.json` que contiene titulares de noticias publicados entre 2012 y 2022 y su categoría.

Campos principales:

campo	descripción
<code>headline</code>	texto del titular
<code>category</code>	etiqueta objetivo (p. ej. U.S. NEWS)

Ejemplo de los datos:

headline	category
Over 4 Million Americans Roll Up Sleeves For Omicron-Targeted COVID Boosters	U.S. NEWS
American Airlines Flyer Charged, Banned For Life After Punching Flight Attendant On Video	U.S. NEWS
23 Of The Funniest Tweets About Cats And Dogs This Week (Sept. 17-23)	COMEDY
The Funniest Tweets From Parents This Week (Sept. 17-23)	PARENTING

headline	category
Woman Who Called Cops On Black Bird-Watcher Loses Lawsuit Against Ex-Employer	U.S. NEWS

Entregables

Un repositorio en GitHub (**por favor crea el repositorio**) con:

- Código fuente en Python.
- README que explique tu proceso y cómo ejecutar todo.
- `requirements.txt` (o equivalente).

No prescribimos estructura de carpetas: organízala como mejor te parezca y justifícalo en el README.

Actividades mínimas

1. Análisis exploratorio (EDA)

- Limpieza de datos, revisión de valores nulos y balance de clases.
- Visualizaciones o tablas que respalden tus hallazgos principales.

2. Preparación y entrenamiento

- Diseña el preprocesamiento necesario para `headline`.
- Entrena **al menos dos** modelos de clasificación distintos.

3. Evaluación

- Reporta métricas típicas de clasificación (accuracy, precision, recall, f1, matriz de confusión).
- Elige un modelo final y explica por qué.

4. API de inferencia

- Implementa una API REST sencilla (Flask o FastAPI).
- Endpoint `POST /predict` que reciba un titular y devuelva la categoría predicha.
- Instrucciones claras en el README para probarla localmente.

Bonus opcionales (elige libremente — no afectan negativamente si los omites)

Bonus	Idea
Modelo LLM	Incluir un enfoque basado en un modelo LLM.
Despliegue en la nube	Publicar la API en un servicio gestionado (p. ej. Cloud Run, App Engine, etc.).
Explainability	Incluir interpretaciones de predicciones.
Monitorización básica	Logs y métricas de rendimiento o calidad del modelo.
Pruebas unitarias	Incorporar tests para las funciones críticas y para el endpoint de predicción.

Criterios de evaluación

criterio	peso
Limpieza y claridad del código	25 %
Calidad del análisis de datos	20 %
Solidez del modelado y justificación	30 %
API funcional y fácil de usar	15 %
Buenas prácticas de entrega y documentación	10 %