**UNIVERSIDAD DE LOS ANDES DEPARTAMENTO DE INGENIERIA DE SISTEMAS Y COMPUTACIÓN**

****

**PROYECTO 1**

**ETAPA 2**

**INTELIGENCIA DE NEGOCIOS**

**FELIPE JAIMES - 202213793**

**MARCOS ESPAÑA - 202124714**

**JAIRO FIERRO - 202226326**

**2025-10**

Contenido

[1.Introducción 3](#_Toc194133944)

[2.Sección 1 3](#_Toc194133945)

[2.1 Primer Endpoint 3](#_Toc194133946)

[2.2 Segundo Endpoint 3](#_Toc194133947)

[2.2.1 Definiciones de reentrenamiento 3](#_Toc194133948)

[2.2.1.1 Reentrenamiento total (batch) 3](#_Toc194133949)

[2.2.1.2 Reentrenamiento incremental (online learning) 3](#_Toc194133950)

[2.2.1.3 Reentrenamiento periódico por validación 4](#_Toc194133951)

[3.Sección 2 4](#_Toc194133952)

[3.1 Descripción del rol del usuario 4](#_Toc194133953)

[3.2 Preguntas respecto a la aplicación 4](#_Toc194133954)

### 1.Introducción

### 2.Sección 1

### 2.1 Primer Endpoint

Este endpoint recibe una o mas entradas de noticias con titulo y descripción para poder hacer una predicción donde se dice si la noticia es falsa o no y también se dice la probabilidad de la predicción.

### 2.2 Segundo Endpoint

### 2.2.1 Definiciones de reentrenamiento

### 2.2.1.1 Reentrenamiento total (batch)

* Descripción: Se entrena un nuevo modelo desde cero utilizando la totalidad del dataset disponible, es decir, la combinación de los datos históricos (viejos) y los datos recién adquiridos (nuevos).
* Ventaja: Se asegura que el modelo no olvida los datos del pasado y captura patrones globales de todo el conjunto de datos, ofreciendo un aprendizaje completo y consistente.
* Desventaja: Costoso en términos de tiempo y recursos computacionales, especialmente cuando el volumen de datos crece significativamente, lo que puede hacerlo impracticable en sistemas con datasets masivos.

### 2.2.1.2 Reentrenamiento incremental (online learning)

* Descripción: Se actualiza el modelo existente incorporando únicamente los nuevos datos a medida que llegan, sin necesidad de reentrenar desde cero con el dataset completo.
* Ventaja: Muy eficiente y rápido, ya que evita procesar nuevamente los datos antiguos. Es ideal para escenarios con datos en streaming o sistemas que requieren adaptarse en tiempo real.
* Desventaja: No todos los modelos o algoritmos soportan aprendizaje incremental de forma nativa, y puede haber riesgo de olvidar patrones importantes de los datos históricos (catastrophic forgetting) si no se implementa cuidadosamente.

### 2.2.1.3 Reentrenamiento periódico por validación

* Descripción: Se reentrena el modelo de forma programada (según un cronograma fijo o cuando una métrica de rendimiento cae por debajo de un umbral predefinido), evaluando previamente si es necesario en función de su desempeño actual.
* Ventaja: Reduce el reentrenamiento innecesario al realizarlo solo cuando hay evidencia de deterioro en el modelo, optimizando recursos y tiempo.
* Desventaja: Existe el riesgo de dejar pasar datos nuevos relevantes que podrían mejorar el modelo antes de que se detecte una caída en el rendimiento, lo que podría retrasar su actualización óptima.

### 3.Sección 2

### 3.1 Descripción del rol del usuario

**Usuario/Rol Principal**

La aplicación está diseñada para Analistas de contenido y verificadores de datos en:

* **Medios de comunicación**: Equipos editoriales que necesitan evaluar noticias políticas antes de publicarlas.
* **Organizaciones de fact-checking**: Profesionales que monitorean información en redes sociales y medios.
* **Instituciones públicas**: Oficinas de prensa o equipos de comunicación que gestionan crisis de desinformación.

**Conexión con Procesos de Negocio**

**Flujo de trabajo actual sin la aplicación:**

* Verificación manual de noticias (lenta y subjetiva).
* Dependencia total del criterio humano para priorizar qué contenido revisar.
* Dificultad para manejar grandes volúmenes de información en tiempos de crisis (elecciones, protestas, etc.).

**Cómo la aplicación se integra en estos procesos:**

**Entrada de datos**:

* Los usuarios ingresan noticias sospechosas mediante una interfaz web o API REST.
* El sistema procesa título + descripción y devuelve una predicción ("verdadera"/"falsa") con su probabilidad asociada.

**Priorización manual mejorada**:

* Los analistas usan las probabilidades para decidir qué noticias requieren revisión urgente (ej.: >80% de ser falsas).
* Noticias con baja probabilidad (<30%) pueden descartarse rápidamente.

**Reentrenamiento opcional**:

* Los usuarios pueden mejorar el modelo añadiendo nuevos casos verificados (vía endpoint /retrain).

**Limitaciones reales de que no es un sistema autónomo:**

* **No toma decisiones automáticas**: Las acciones (bloquear, desmentir, etc.) dependen del usuario.
* **No está integrado directamente con CMS o redes sociales**: Requiere intervención manual para importar/exportar datos.
* **No genera alertas proactivas**: Los usuarios deben consultar la herramienta activamente.

**Importancia para el Rol**

Esta aplicación es crucial para los analistas y verificadores porque:

**1. Agiliza el Proceso de Verificación**

* Análisis inmediato al procesar noticias em formato de título + descripción al instante, dando resultados con probabilidades claras.
* Priorización eficiente ya que los usuarios pueden enfocarse en las noticias más sospechosas como >80% de probabilidad de ser falsas y descartar las menos riesgosas <30%.

**2. Reduce la Carga de Trabajo Manual**

* Automatiza la primera revisión, evitando que los equipos pierdan tiempo en noticias obviamente verdaderas o falsas.
* Permite manejar grandes volúmenes de datos, algo imposible con métodos puramente manuales.

**3. Proporciona Consistencia en la Toma de Decisiones**

* Evita sesgos humanos ya que el modelo aplica los mismos criterios a todas las noticias, sin influencia de fatiga o subjetividad.
* Da una base objetiva para justificar por qué una noticia se considera falsa (probabilidad + palabras clave detectadas).

**4. Facilita la Mejora Continua**

* Reentrenamiento con nuevos datos porque a medida que los verificadores añaden casos confirmados, el modelo se vuelve más preciso en contextos específicos.
* Da métricas claras para que los usuarios pueden monitorear el rendimiento del modelo para asegurar su confiabilidad.

**5. Prepara el Terreno para Integraciones Futuras**

* API disponible ,aunque hoy funciona como herramienta independiente, su diseño permite conectar con CMS, redes sociales o sistemas de alerta cuando sea necesario.

### 3.2 Preguntas respecto a la aplicación

**1. Recursos Informáticos Requeridos**

**Justificación de Requerimientos:**  
El diseño actual priorizó eficiencia sobre capacidad escalable, considerando:

* **Modelo Ligero (8.4MB):**
  + Uso de RandomForest (no profundo) con TF-IDF para NLP, optimizando memoria.
  + Serialización con joblib que es más eficiente que pickle para objetos scikit-learn.
* **Infraestructura Mínima:**
  + 1 CPU/2GB RAM suficientes para inferencia, ya que:
    - La vectorización TF-IDF ocurre en memoria.
    - FastAPI + Uvicorn tienen bajo overhead.
  + 10MB de almacenamiento cubren:
    - Modelo serializado + datos históricos (sin logs extensos).

**2. Integración Organizacional**

**Decisiones de Implementación:**

* **API REST Simple (FastAPI):**
  + Se eligió por su balance entre rendimiento y facilidad de desarrollo.
  + Endpoints minimalistas (/predict, /retrain) para:
    - Evitar complejidad prematura.
    - Permitir extensiones futuras (ej.: /monitor).
* **Interfaz Web Básica:**
  + Propósito: Demostrar funcionalidad sin depender de integraciones.
  + Limitación: No hay autenticación, pensada para uso interno inicial.

**Flujo de Trabajo Habilitado:**

1. **Medios de Comunicación:**
   * Copiar/pegar noticias en la web → obtener predicciones → decidir manualmente.
2. **Equipos de Fact-Checking:**
   * Reentrenar modelo semanalmente con nuevos casos (vía curl o interfaz).
3. **Gobiernos:**
   * Consumir API desde scripts internos (ej.: monitoreo de redes sociales manual).

**Barreras Actuales:**

* La falta de webhooks/exportación obliga a procesos manuales de extracción de datos.
* No hay conectores preconstruidos para CMS (WordPress) o redes sociales.

**3. Riesgos Operacionales**

**Análisis de Decisiones Técnicas:**

1. **CORS Permisivo:**
   * Se implementó así para facilitar pruebas iniciales, pero expone a:
     + CSRF si se despliega en Internet sin firewall.
     + Uso no autorizado de la API.
2. **Ausencia de Umbrales:**
   * El código devuelve probabilidades "crudas", porque:
     + Los umbrales óptimos varían por organización (ej.: medios vs. gobiernos).
     + Requeriría una capa de configuración no priorizada en el MVP.
3. **Falta de Logging:**
   * Omitido para simplificar el despliegue inicial, pero:
     + Dificulta diagnosticar errores en producción.
     + Impide auditoría de decisiones automatizadas.