Documentación Técnica: Sistema Predictivo de Agendamiento CESFAM

Proyecto M - Minería de Datos

- **Versión:** 1.2.0
- **Fecha:** noviembre 2025

Contenido

1. Definición del Problema y Solución

1.1 Contexto del Problema

Los Centros de Salud Familiar (CESFAM) enfrentan una problemática crítica relacionada con la eficiencia en la gestión de horas médicas. Actualmente, existe una alta tasa de **inasistencia (No-Show)**, donde los pacientes agendan horas pero no se presentan, generando tiempos ociosos para los médicos y aumentando las listas de espera para otros pacientes.

1.2 Impacto en el Negocio

- **Pérdida de recursos:** Horas profesionales pagadas pero no utilizadas.
- **Ineficiencia:** Dificultad para reasignar cupos de forma reactiva.
- **Salud Pública:** Retraso en la atención de pacientes que realmente lo necesitan.

1.3 Solución Propuesta

Se ha desarrollado un sistema inteligente basado en **Machine Learning** que predice la probabilidad de que un paciente falte a su cita. Esta predicción permite al CESFAM tomar medidas proactivas, como el sobre-agendamiento controlado o el envío de recordatorios personalizados.

2. Arquitectura del Sistema

El proyecto sigue una arquitectura modular de microservicios:

1. **Ingesta de Datos:** Script generador de datos sintéticos basado en patrones demográficos reales (data_generator.py).
2. **Pipeline de ML:** Procesamiento, limpieza y entrenamiento (pipeline.py, train.py).
3. **Modelo Serializado:** Archivo binario (model_pipeline.pkl) que contiene el modelo entrenado.
4. **API REST:** Microservicio desarrollado en **FastAPI** que expone el modelo (main.py).
5. **Frontend:** Dashboard interactivo en **Streamlit** para consumo del usuario final (dashboard.py).

3. Datos y Análisis Exploratorio (EDA)

3.1 Fuente de Datos

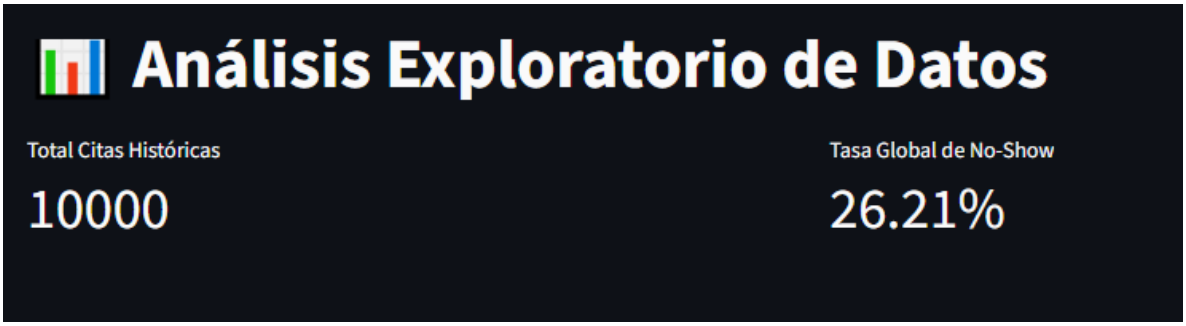
Se utilizó un dataset sintético de 10,000 registros diseñado para simular el comportamiento real de pacientes en Chile.

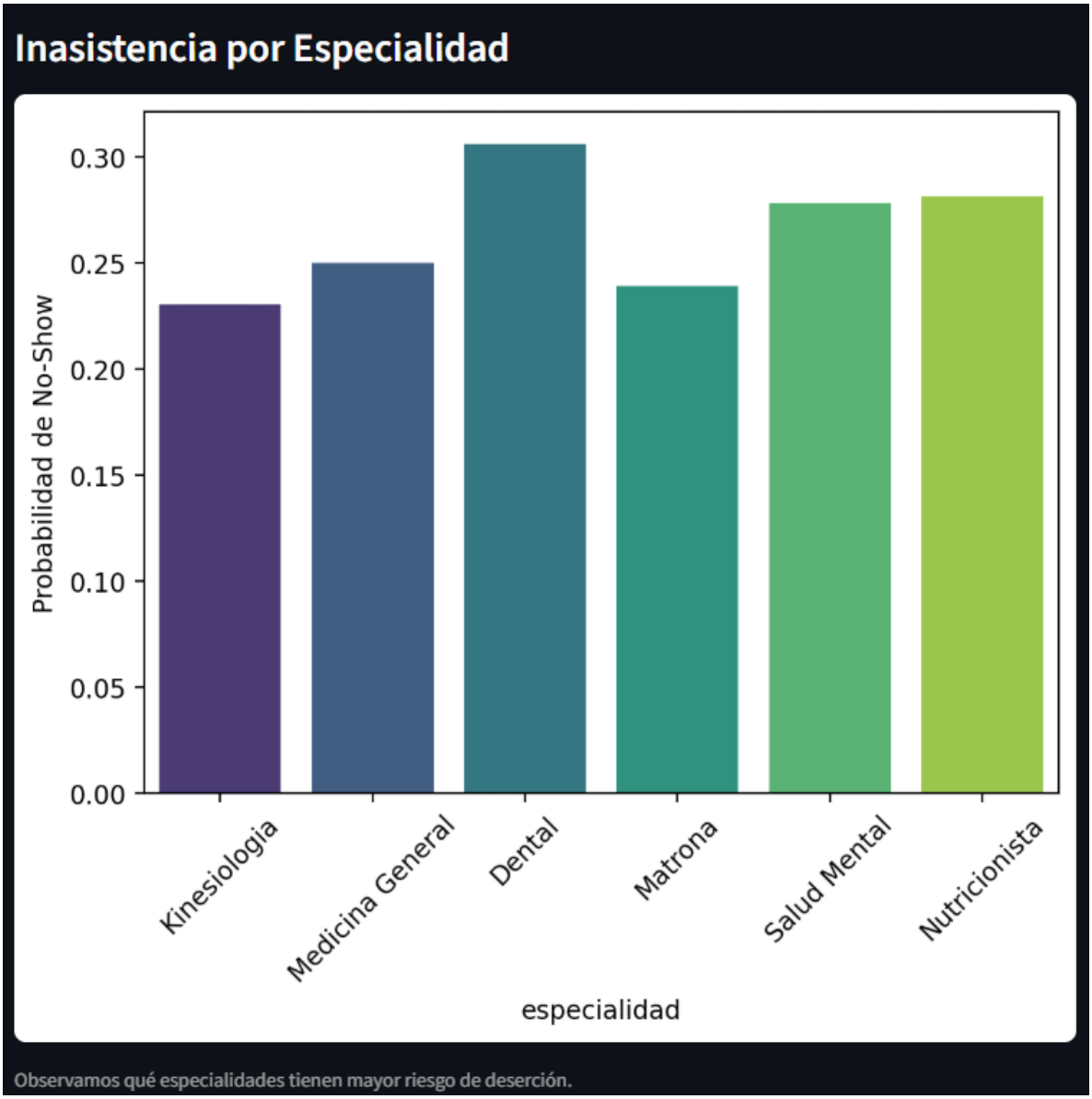
- **Ubicación:** data/dataset_cesfam_v1.csv

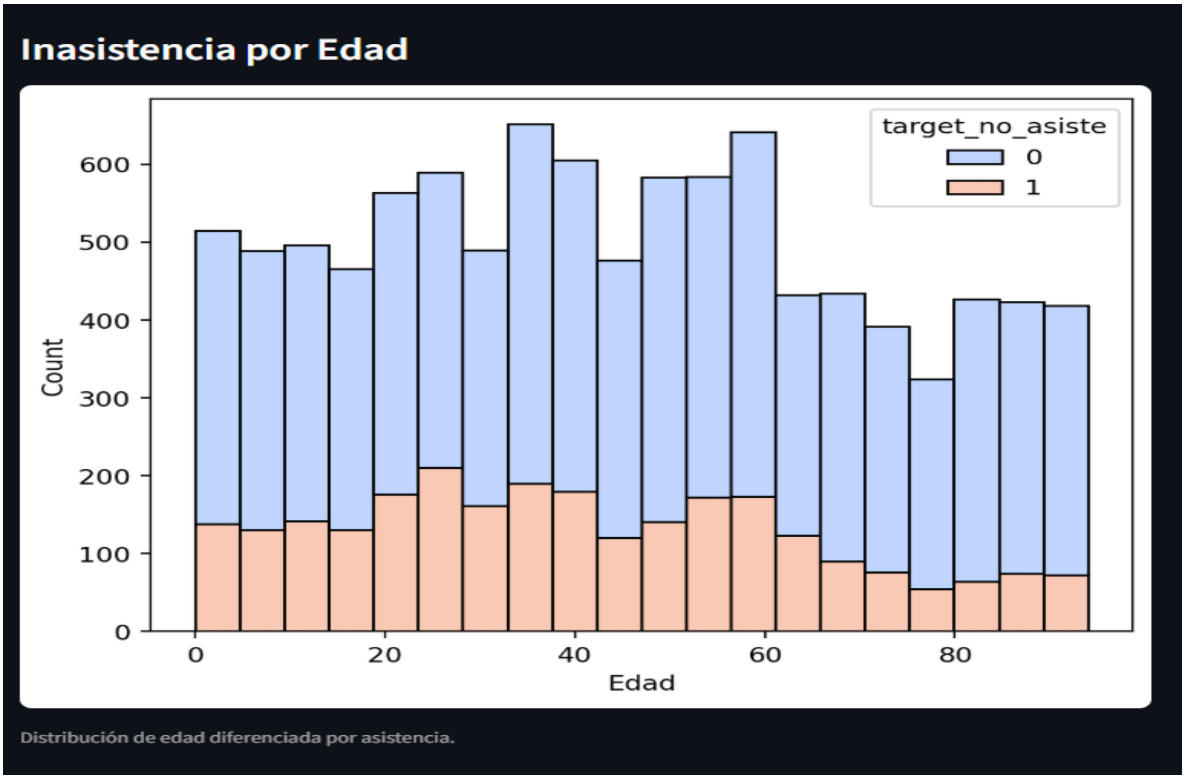
3.2 Diccionario de Datos (Variables)

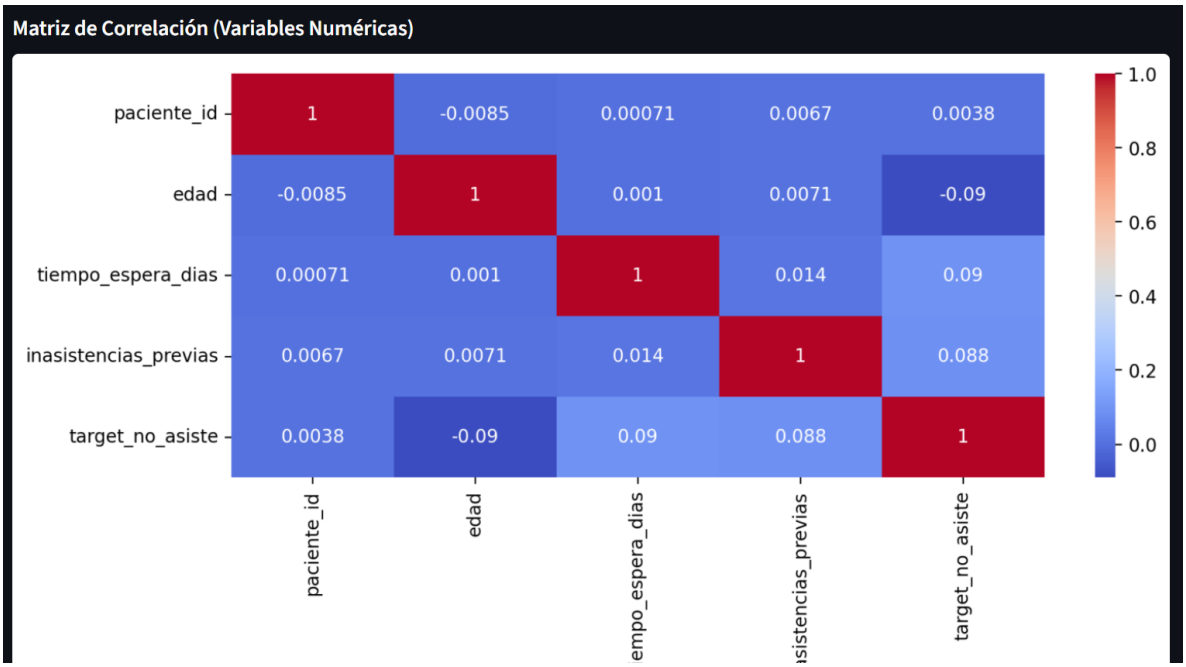
Variable	**Tipo**	**Descripción**
edad	Numérica	Edad del paciente (0-95 años).
sexo	Categórica	Género del paciente.
sector	Categórica	Sector geográfico (Norte, Sur, Centro, Rural).
prevision	Categórica	Tramo de Fonasa (A, B, C, D).
especialidad	Categórica	Tipo de atención (Dental, Medicina General, etc.).
tiempo_espera_dias	Numérica	Días transcurridos entre la solicitud y la cita.
inasistencias_previas	Numérica	Cantidad histórica de faltas del paciente.
target_no_asiste	**Binaria**	**Variable objetivo (1: No asiste, 0: Asiste).**

3.3 Hallazgos del EDA (Insights)









- **Patrón de Historial:** Se observa una fuerte correlación positiva entre inasistencias_previas y la probabilidad de volver a faltar.
- **Factor Tiempo:** Las citas agendadas con más de 20 días de anticipación muestran una mayor tasa de deserción (olvido).
- **Edad:** Los segmentos de adultos jóvenes (20-35 años) presentan mayor riesgo de inasistencia comparado con adultos mayores.

4. Preprocesamiento e Ingeniería de Características

Se implementó un Pipeline automatizado con scikit-learn para garantizar la reproducibilidad en producción.

4.1 Manejo de Nulos

- **Variables Numéricas:** Imputación utilizando la **Mediana** (SimpleImputer).
- **Variables Categóricas:** Imputación utilizando la **Moda** (valor más frecuente).

4.2 Codificación (Encoding)

- Se aplicó **One-Hot Encoding** a las variables categóricas (sexo, sector, especialidad, prevision, turno, dia_semana).
- Se configuró handle_unknown='ignore' para asegurar que la API no falle si recibe categorías nuevas en el futuro.

4.3 Escalado

- Se aplicó **StandardScaler** a las variables numéricas para normalizar la distribución de los datos.

5. Modelo Predictivo

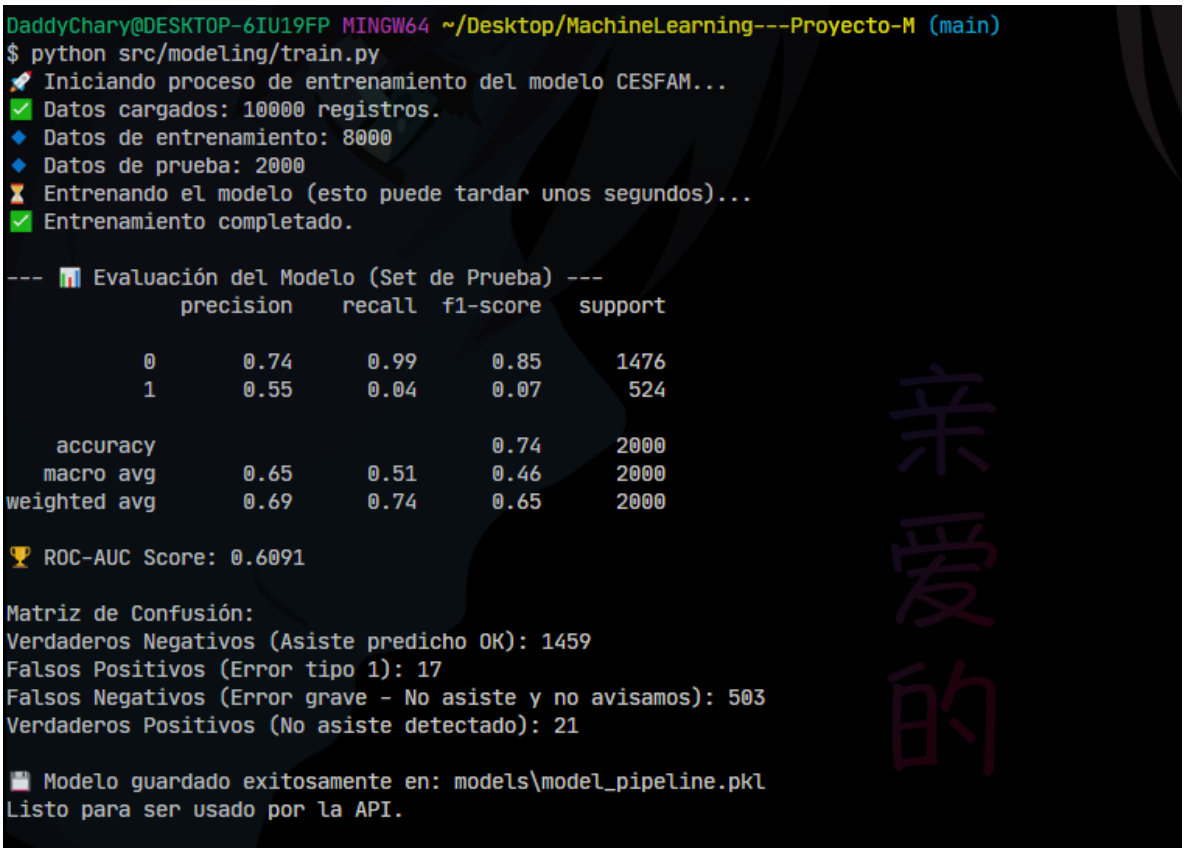
5.1 Algoritmo Seleccionado

- **Modelo:** Gradient Boosting Classifier.
- **Justificación:** Este algoritmo maneja eficazmente relaciones no lineales y es robusto frente a datos tabulares desbalanceados, superando generalmente a modelos lineales simples.

5.2 Entrenamiento y Validación

- **Split:** 80% Entrenamiento / 20% Prueba (Estratificado).
- **Hiperparámetros:**
    - n_estimators: 100
    - learning_rate: 0.1
    - max_depth: 3

5.3 Métricas de Desempeño



6. API y Consumo del Modelo

6.1 Endpoint /predict

El sistema expone una API RESTful documentada automáticamente con Swagger.

Ejemplo de Request (JSON):

JSON

```
{
  "edad": 30,
  "sexo": "Femenino",
  "sector": "Norte",
  "prevision": "Fonasa B",
  "especialidad": "Dental",
  "dia_semana": "Lunes",
  "turno": "Mañana",
  "tiempo_espera_dias": 5,
  "inasistencias_previas": 2
}
```

Respuesta:

JSON

```
{
  "prediccion": 1,
  "probabilidad": 0.85,
  "mensaje": "Alto riesgo de inasistencia"
}
```

7. Pruebas y Evidencias

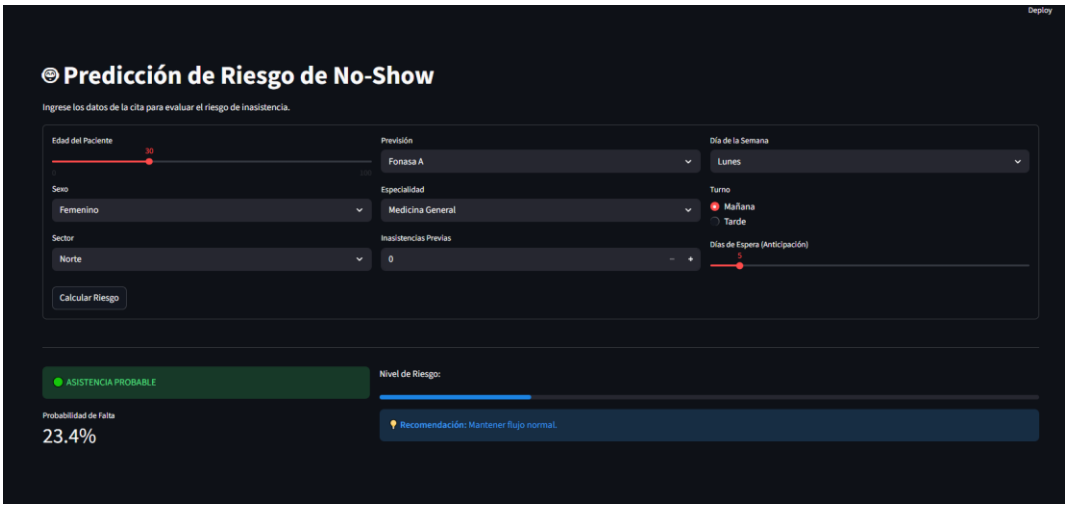
7.1 Tests Unitarios

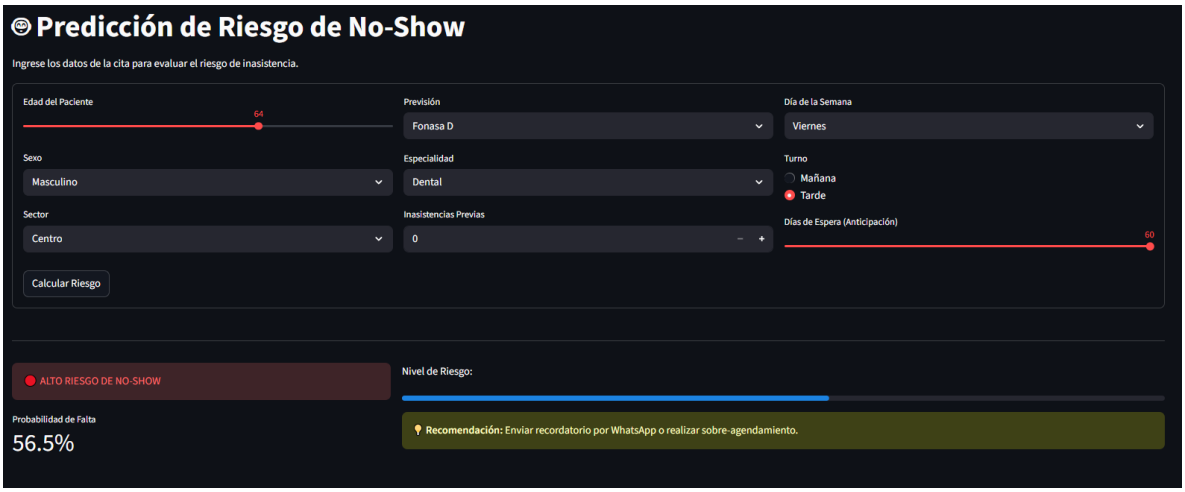
Se ejecutaron pruebas unitarias automatizadas (tests/) validando:

1. Creación correcta del Pipeline.
2. Imputación automática de valores nulos.
3. Disponibilidad del endpoint de la API.

7.2 Pruebas Funcionales

1. **Formulario de entrada:** Datos ingresados por el usuario.
2. **Predicción exitosa:** Visualización del resultado (Rojo/Verde).





8. Conclusiones

El **Proyecto M** demuestra la viabilidad técnica de utilizar inteligencia artificial para optimizar la gestión de recursos en salud pública. La solución implementada es modular, escalable y cuenta con una interfaz amigable para el usuario final, cumpliendo con todos los requisitos de ingeniería de software y ciencia de datos planteados.