

Documentación Técnica: Sistema Predictivo de Agendamiento CESFAM

Proyecto M - Minería de Datos

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

Contenido

Documentación Técnica: Sistema Predictivo de Agendamiento CESFAM	1
1. Definición del Problema y Solución	3
1.1 Contexto del Problema	3
1.2 Impacto en el Negocio.....	3
1.3 Solución Propuesta	3
2. Arquitectura del Sistema	3
3. Datos y Análisis Exploratorio (EDA)	4
3.1 Fuente de Datos.....	4
3.2 Diccionario de Datos (Variables)	4
3.3 Hallazgos del EDA (Insights)	5
4. Preprocesamiento e Ingeniería de Características.....	7
4.1 Manejo de Nulos	7
4.2 Codificación (Encoding).....	7
4.3 Escalado	7
5. Modelo Predictivo	8
5.1 Algoritmo Seleccionado.....	8
5.2 Entrenamiento y Validación.....	8
5.3 Métricas de Desempeño.....	8
6. API y Consumo del Modelo.....	9
6.1 Endpoint /predict	9
7. Pruebas y Evidencias	10
7.1 Tests Unitarios	10
7.2 Pruebas Funcionales.....	11
8. Conclusiones	12

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)



Análisis Exploratorio de Datos

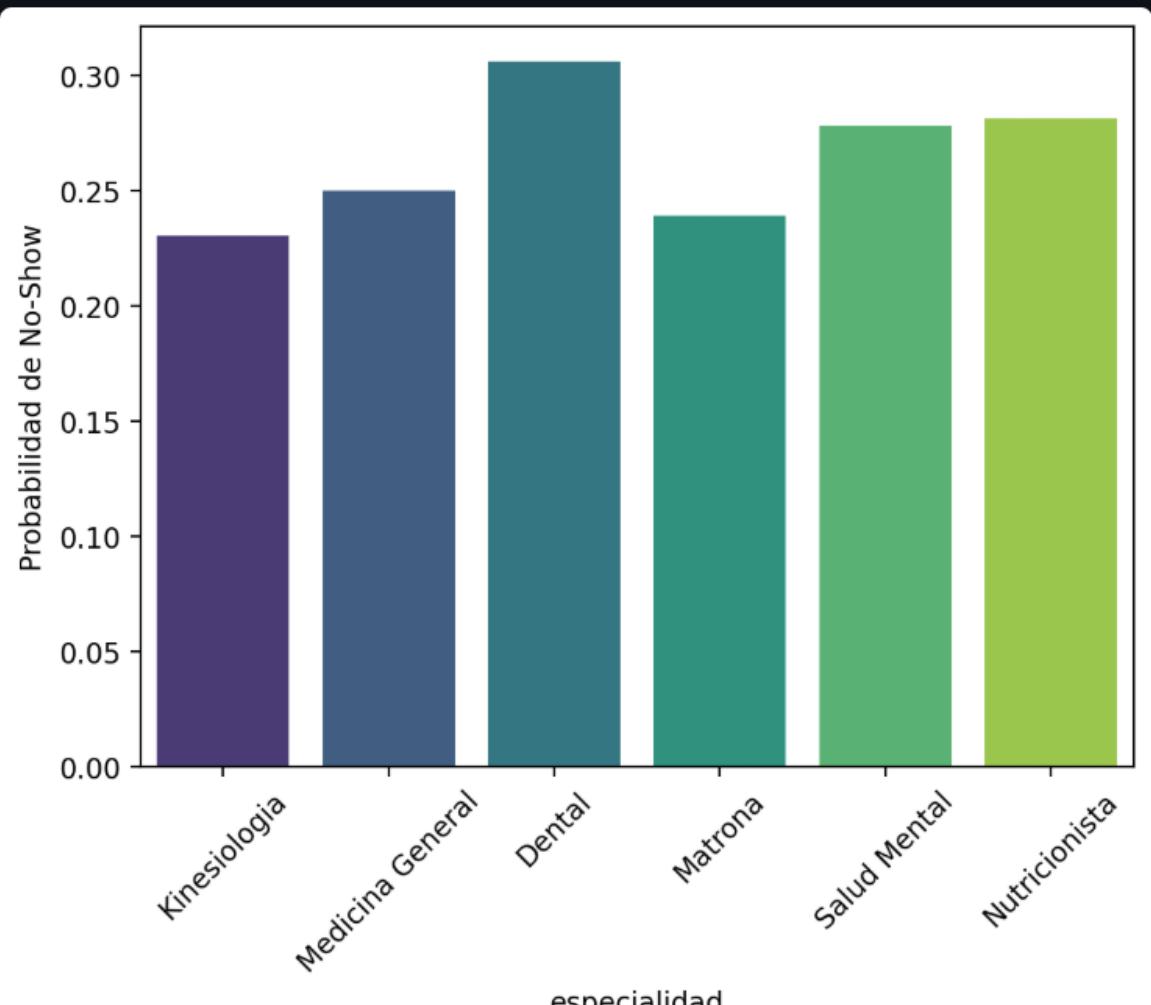
Total Citas Históricas

10000

Tasa Global de No-Show

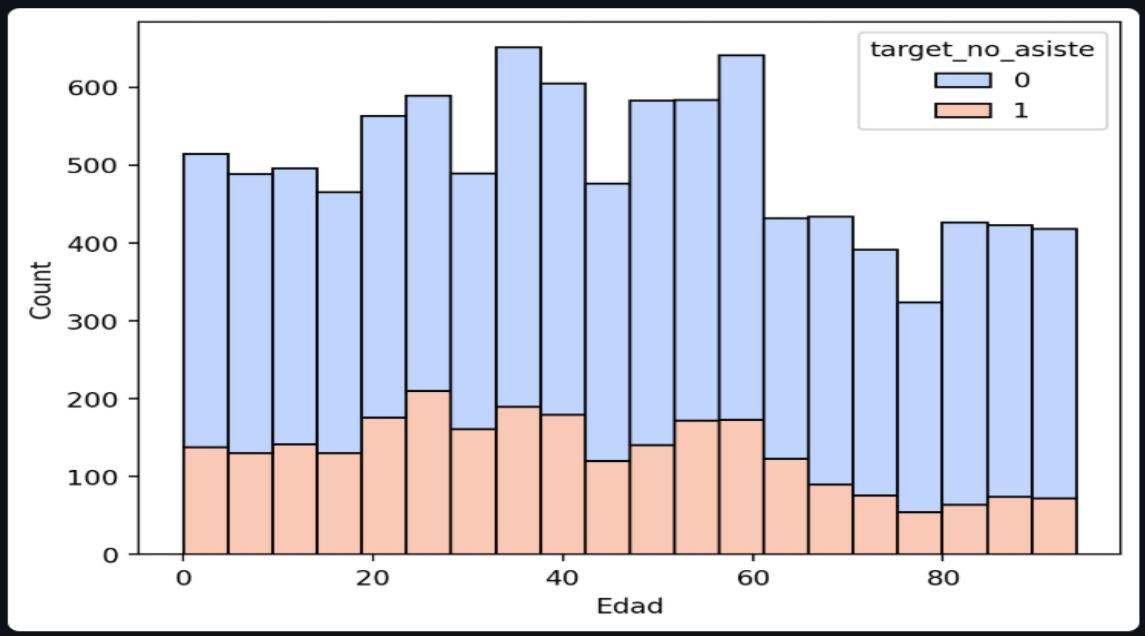
26.21%

Inasistencia por Especialidad



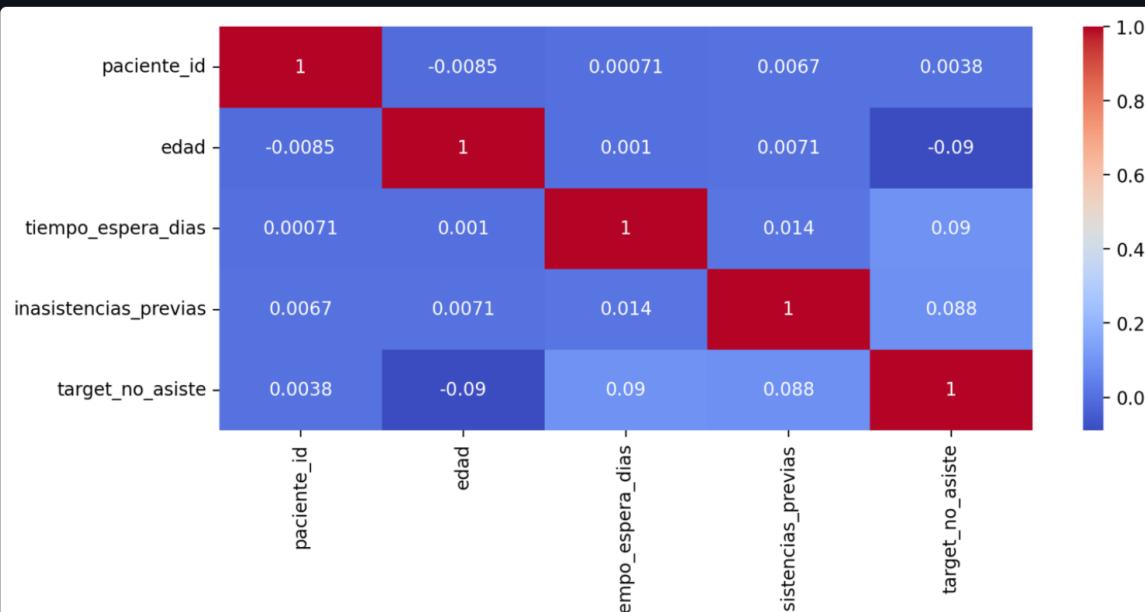
Observamos qué especialidades tienen mayor riesgo de deserción.

Inasistencia por Edad



Distribución de edad diferenciada por asistencia.

Matriz de Correlación (Variables Numéricas)



- 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

```
DaddyChary@DESKTOP-6IU19FP MINGW64 ~/Desktop/MachineLearning---Proyecto-M (main)
$ python src/modeling/train.py
⚡ Iniciando proceso de entrenamiento del modelo CESFAM...
✓ Datos cargados: 10000 registros.
◆ Datos de entrenamiento: 8000
◆ Datos de prueba: 2000
⏳ Entrenando el modelo (esto puede tardar unos segundos)...
✓ Entrenamiento completado.

--- 📈 Evaluación del Modelo (Set de Prueba) ---
      precision    recall   f1-score   support
          0       0.74      0.99      0.85     1476
          1       0.55      0.04      0.07      524

      accuracy                           0.74     2000
      macro avg       0.65      0.51      0.46     2000
  weighted avg       0.69      0.74      0.65     2000

🏆 ROC-AUC Score: 0.6091

Matriz de Confusión:
Verdaderos Negativos (Asiste predicho OK): 1459
Falsos Positivos (Error tipo 1): 17
Falsos Negativos (Error grave - No asiste y no avisamos): 503
Verdaderos Positivos (No asiste detectado): 21

💾 Modelo guardado exitosamente en: models\model_pipeline.pkl
Listo para ser usado por la API.
```

亲爱的

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.

```
$ python src/data_prep/data_generator.py
Generando 10000 registros sintéticos...
Dataset guardado exitosamente en: data/dataset_cesfam_v1.csv
Tasa de inasistencia simulada: 26.21%

DaddyChary@DESKTOP-6IU19FP MINGW64 ~/Desktop/MachineLearning---Proyecto-M (main)
$ python -m unittest tests/test_preprocess.py
.c:\Users\Da...Python\site-packages\sklearn\impute\_base.py:653: UserWarning: Skipping features without
any observed values: ['edad']. At least one non-missing value is needed for imputation with strategy='median'.
warnings.warn(
c:\Users\Da...Python\site-packages\sklearn\impute\_base.py:653: UserWarning: Skipping features without
any observed values: ['sexo']. At least one non-missing value is needed for imputation with strategy='most_frequent'.
warnings.warn(
    El pipeline manejo correctamente los valores Nulos (NaN).
...
    El pipeline manejo correctamente una categoría desconocida.
.

Ran 4 tests in 0.055s
OK
.DaddyChary@DESKTOP-6IU19FP MINGW64 ~/Desktop/MachineLearning---Proyecto-M (main)
$
```

7.2 Pruebas Funcionales

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

The screenshot shows the application's interface for predicting no-show risk. The form fields are as follows:

- Edad del Paciente: 30
- Previsión: Fonasa A
- Día de la Semana: Lunes
- Sexo: Femenino
- Especialidad: Medicina General
- Turno: Mañana (selected)
- Sector: Norte
- Inasistencias Previas: 0
- Días de Espera (Anticipación): 0

Below the form, a green bar indicates "ASISTENCIA PROBABLE" (Probable Attendance) with a probability of 23.4%.

The screenshot shows the application's interface for predicting no-show risk. The form fields are as follows:

- Edad del Paciente: 64
- Previsión: Fonasa D
- Día de la Semana: Viernes
- Sexo: Masculino
- Especialidad: Dental
- Turno: Tarde (selected)
- Sector: Centro
- Inasistencias Previas: 0
- Días de Espera (Anticipación): 60

Below the form, a red bar indicates "ALTO RIESGO DE NO-SHOW" (High Risk of No-Show) with a probability of 56.5%.

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.