



# Manual de Usuario

## Mortality AMI Predictor

Sistema de Predicción de Mortalidad  
en Infarto Agudo de Miocardio



Personal Clínico



Científicos de Datos



Desarrolladores

Versión 2.0 | Diciembre 2025

# Índice

|  |          |
|--|----------|
| <b>1. Introducción</b>   | <b>4</b> |
| 1.1. Bienvenido a Mortality AMI Predictor . . . . .              | 4        |
| 1.2. Perfiles de Usuario . . . . .                               | 4        |
| <b>2. Inicio Rápido</b>  | <b>5</b> |
| 2.1. Requisitos del Sistema . . . . .                            | 5        |
| 2.2. Instalación . . . . .                                       | 6        |
| 2.3. Primera Vista del Dashboard . . . . .                       | 6        |
| <b>3. Guía para Personal Clínico</b>                             | <b>6</b> |
| 3.1. Calculadora de Scores Clínicos . . . . .                    | 7        |
| 3.1.1. Score GRACE . . . . .                                     | 7        |
| 3.1.2. Score TIMI . . . . .                                      | 7        |
| 3.2. Realizar Predicciones . . . . .                             | 8        |
| 3.2.1. Predicción Individual . . . . .                           | 8        |
| 3.3. Entender las Predicciones (Explicabilidad Básica) . . . . . | 8        |
| 3.3.1. Gráfico de Importancia de Variables . . . . .             | 9        |
| 3.3.2. Ejemplo de Interpretación . . . . .                       | 9        |
| <b>4. Guía para Científicos de Datos</b>                         | <b>9</b> |
| 4.1. Carga y Limpieza de Datos . . . . .                         | 9        |
| 4.1.1. Carga de Datos . . . . .                                  | 9        |
| 4.1.2. Pipeline de Limpieza . . . . .                            | 10       |
| 4.2. Análisis Exploratorio de Datos (EDA) . . . . .              | 10       |
| 4.2.1. Análisis Univariado . . . . .                             | 10       |
| 4.2.2. Análisis Bivariado . . . . .                              | 11       |
| 4.2.3. Análisis Multivariado . . . . .                           | 11       |
| 4.3. Entrenamiento de Modelos . . . . .                          | 11       |
| 4.3.1. Algoritmos Disponibles . . . . .                          | 11       |

---

|  |           |
|--|-----------|
| 4.3.2. Validación Cruzada Rigurosa . . . . .   | 11        |
| 4.4. Evaluación de Modelos . . . . .           | 12        |
| 4.4.1. Métricas de Clasificación . . . . .     | 12        |
| 4.4.2. Curvas de Calibración . . . . .         | 12        |
| 4.4.3. Decision Curve Analysis (DCA) . . . . . | 13        |
| 4.5. Explicabilidad Avanzada (SHAP) . . . . .  | 13        |
| 4.5.1. Visualizaciones SHAP . . . . .          | 13        |
| 4.5.2. Interpretación de SHAP . . . . .        | 13        |
| 4.6. AutoML . . . . .                          | 13        |
| 4.6.1. Backends Disponibles . . . . .          | 14        |
| 4.6.2. Presets de Búsqueda . . . . .           | 14        |
| 4.7. Optimización Inversa . . . . .            | 14        |
| 4.7.1. Concepto . . . . .                      | 14        |
| 4.7.2. Configuración . . . . .                 | 14        |
| <b>5. Guía para Desarrolladores</b>            | <b>14</b> |
| 5.1. Modelos Personalizados . . . . .          | 15        |
| 5.1.1. Arquitectura Base . . . . .             | 15        |
| 5.1.2. Métodos Requeridos . . . . .            | 15        |
| 5.1.3. Flujo de Trabajo . . . . .              | 16        |
| 5.2. Despliegue con Docker . . . . .           | 16        |
| 5.2.1. Construcción de la Imagen . . . . .     | 16        |
| 5.2.2. Ejecución del Contenedor . . . . .      | 16        |
| 5.2.3. Docker Compose . . . . .                | 16        |
| 5.3. Estructura del Código . . . . .           | 17        |
| 5.4. API Programática . . . . .                | 17        |
| <b>6. Referencia de Páginas del Dashboard</b>  | <b>18</b> |
| <b>7. Solución de Problemas</b>                | <b>18</b> |
| 7.1. Problemas Comunes . . . . .               | 18        |

---

|   |           |
|---|-----------|
| 7.2. Mensajes de Error Frecuentes . . . . . | 19        |
| <b>8. Glosario</b>                          | <b>19</b> |
| <b>9. Contacto y Soporte</b>                | <b>20</b> |

## 1 Introducción

### 1.1 Bienvenido a Mortality AMI Predictor

**Mortality AMI Predictor** es una aplicación integral para la predicción de mortalidad intrahospitalaria en pacientes con Infarto Agudo de Miocardio (IAM). El sistema combina técnicas avanzadas de Machine Learning con una interfaz intuitiva diseñada para diferentes perfiles de usuario.

#### Propósito del Sistema

El objetivo principal es proporcionar herramientas que ayuden a:

- Predecir el riesgo de mortalidad intrahospitalaria
- Calcular scores clínicos validados (GRACE, TIMI)
- Analizar factores de riesgo mediante explicabilidad
- Optimizar intervenciones mediante análisis “what-if”

#### Uso Responsable

Esta herramienta es para **finés de investigación y educación**. Las decisiones clínicas deben ser tomadas siempre por profesionales de salud cualificados utilizando su juicio clínico.

### 1.2 Perfiles de Usuario

El sistema está diseñado para tres tipos principales de usuarios, cada uno con diferentes necesidades y niveles de experiencia técnica:



#### Personal Clínico

**Médicos, enfermeros y personal sanitario** que necesitan:

- Calcular scores de riesgo clínico rápidamente
- Obtener predicciones de mortalidad para pacientes
- Entender qué factores influyen en el pronóstico

**Secciones recomendadas:** Predicciones, Scores Clínicos, Explicabilidad básica

### Científicos de Datos / Investigadores

**Analistas y estadísticos** que requieren:

- Análisis exploratorio de datos clínicos
- Entrenamiento y comparación de modelos
- Evaluación rigurosa con métricas estándar
- Análisis de explicabilidad avanzado (SHAP)

**Secciones recomendadas:** Limpieza de Datos, EDA, Entrenamiento, Evaluación, AutoML

### Desarrolladores / Programadores

**Ingenieros de ML y desarrolladores** que buscan:

- Crear modelos personalizados
- Integrar con sistemas externos
- Extender funcionalidades
- Desplegar en producción con Docker

**Secciones recomendadas:** Modelos Personalizados, API, Despliegue Docker

## 2 Inicio Rápido

### 2.1 Requisitos del Sistema

| Componente        | Requisito  |
|-------------------|--|
| Sistema Operativo | Windows 10/11, macOS 10.15+, Linux (Ubuntu 20.04+)   |
| Python            | 3.9 o superior                                       |
| RAM               | Mínimo 8 GB (16 GB recomendado para AutoML)          |
| Espacio en disco  | 2 GB para instalación + espacio para datasets        |
| Navegador         | Chrome, Firefox, Edge o Safari (versiones recientes) |

Cuadro 1: Requisitos mínimos del sistema

## 2.2 Instalación

**Paso 1:** Clonar el repositorio o descargar los archivos del proyecto:

```
git clone https://github.com/tu-usuario/mortality-ami-predictor.git
cd mortality-ami-predictor/Tools
```

**Paso 2:** Crear un entorno virtual e instalar dependencias:

```
python -m venv venv
venv\Scripts\activate      # Windows
source venv/bin/activate   # Linux/macOS
pip install -r requirements.txt
```

**Paso 3:** Iniciar la aplicación:





```
streamlit run dashboard/Dashboard.py
```

✓ Listo

La aplicación se abrirá automáticamente en tu navegador en <http://localhost:8501>

## 2.3 Primera Vista del Dashboard

Al iniciar la aplicación, verás la página principal con:

-  **Logo y título** del sistema
-  **Barra lateral** con navegación a todas las páginas
-  **Estadísticas rápidas** del dataset (si está configurado)
-  **Información** sobre las funcionalidades disponibles


## 3 Guía para Personal Clínico



### Sección para Personal Clínico

Esta sección está diseñada para médicos y personal sanitario.  
No requiere conocimientos de programación ni estadística avanzada.


### 3.1 Calculadora de Scores Clínicos

La página  **Clinical Scores** permite calcular los scores de riesgo más utilizados en cardiología.

#### 3.1.1 Score GRACE

El **GRACE Score** (Global Registry of Acute Coronary Events) estima la mortalidad intrahospitalaria y a 6 meses en pacientes con síndrome coronario agudo.

**Para calcular el score GRACE:**

1. Navega a **06**  **Clinical Scores**
2. Selecciona “GRACE Score” en el menú desplegable
3. Introduce los datos del paciente:

| Variable                   | Unidad | Descripción                             |
|----------------------------|--------|---|
| Edad                       | años   | Edad del paciente                       |
| Frecuencia cardíaca        | lpm    | Latidos por minuto al ingreso           |
| Presión arterial sistólica | mmHg   | PA sistólica al ingreso                 |
| Creatinina                 | mg/dL  | Nivel sérico de creatinina              |
| Clase Killip               | I-IV   | Clasificación de insuficiencia cardíaca |
| Desviación ST              | Sí/No  | Elevación o depresión del ST en ECG     |
| Enzimas cardíacas          | Sí/No  | Troponina o CK-MB elevadas              |
| Parada cardíaca            | Sí/No  | PCR al ingreso                          |

4. Haz clic en **Calcular GRACE Score**
5. Revisa el resultado con la categoría de riesgo

#### Interpretación del GRACE Score

- **Bajo riesgo:** < 109 puntos (mortalidad < 1 %)
- **Riesgo intermedio:** 109-140 puntos (mortalidad 1-3 %)
- **Alto riesgo:** > 140 puntos (mortalidad > 3 %)

#### 3.1.2 Score TIMI

El **TIMI Score** evalúa el riesgo en pacientes con IAM con elevación del ST.

**Variables del TIMI:**

- Edad  $\geq$  65 años



- Diabetes mellitus
- Hipertensión arterial
- Infarto previo
- Otros factores cardiovasculares

## 3.2 Realizar Predicciones

La página [🔗 Predictions](#) permite obtener predicciones de mortalidad para pacientes individuales.

### 3.2.1 Predicción Individual

1. Navega a [03 🔗 Predictions](#)
2. Selecciona un modelo entrenado en la barra lateral
3. Introduce los valores de las variables clínicas del paciente
4. El sistema mostrará:
  - **Probabilidad de mortalidad** (0-100 %)
  - **Clasificación de riesgo** (bajo/medio/alto)
  - **Factores más influyentes** en la predicción

#### ⚠ Interpretación Clínica

Las predicciones del modelo son **probabilísticas** y deben interpretarse en el contexto clínico completo del paciente. Considere siempre:

- Comorbilidades no incluidas en el modelo
- Factores socioeconómicos
- Preferencias del paciente
- Recursos disponibles

## 3.3 Entender las Predicciones (Explicabilidad Básica)

La página [🔗 Explainability](#) ayuda a entender **por qué** el modelo hace una predicción.

### 3.3.1 Gráfico de Importancia de Variables

Este gráfico muestra qué variables tienen mayor influencia en la predicción:

- **Barras rojas:** Variables que **aumentan** el riesgo
- **Barras verdes:** Variables que **disminuyen** el riesgo
- **Longitud de la barra:** Magnitud del impacto

### 3.3.2 Ejemplo de Interpretación

#### Paciente ejemplo:

- Probabilidad de mortalidad: **35 %**
- Factor principal de riesgo: **Edad avanzada (+12 %)**
- Factor protector principal: **PA sistólica normal (-8 %)**

**Interpretación:** La edad del paciente es el factor que más contribuye al riesgo, mientras que una presión arterial sistólica dentro de rangos normales actúa como factor protector.


## 4 Guía para Científicos de Datos



### Sección para Científicos de Datos

Esta sección asume conocimientos de estadística, ML y análisis de datos.

### 4.1 Carga y Limpieza de Datos

La página  **Data Cleaning and EDA** es el punto de entrada para cualquier análisis.

#### 4.1.1 Carga de Datos

El sistema soporta múltiples formatos y fuentes:

| Método           | Descripción                            | Formatos       |
|------------------|--|----------------|
| Ruta del archivo | Especificar path absoluto o relativo   | CSV, Excel     |
| Subir archivo    | Drag & drop o selección de archivo     | CSV, XLSX, XLS |
| Dataset limpio   | Cargar datasets previamente procesados | CSV            |

### 4.1.2 Pipeline de Limpieza

El módulo `DataCleaner` proporciona un pipeline configurable:

#### 1. Imputación de valores faltantes

- Numéricos: media, mediana, KNN
- Categóricos: moda, valor constante

#### 2. Detección y tratamiento de outliers

- Métodos: IQR, Z-score
- Tratamiento: cap (winsorización), remove, impute

#### 3. Codificación de variables categóricas

- Label encoding
- One-hot encoding
- Target encoding

#### 4. Escalado de características

- `StandardScaler` (z-score)
- `MinMaxScaler`
- `RobustScaler`

#### ✓ Tip: Configuración Recomendada

Para datos clínicos con outliers y valores faltantes:

- Imputación numérica: **mediana** (robusta a outliers)
- Outliers: **IQR** + **cap** (preserva datos)
- Escalado: **RobustScaler** (insensible a outliers)

## 4.2 Análisis Exploratorio de Datos (EDA)

### 4.2.1 Análisis Univariado

- **Variables numéricas:** Histogramas, boxplots, estadísticos descriptivos ( $\mu$ ,  $\sigma$ , mediana, IQR)
- **Variables categóricas:** Gráficos de barras, frecuencias absolutas y relativas

### 4.2.2 Análisis Bivariado

- **Numérico-Numérico:** Matriz de correlación (Pearson, Spearman), scatter plots
- **Numérico-Categorico:** Boxplots por grupo, pruebas t/Mann-Whitney
- **Categorico-Categorico:** Tablas de contingencia, Chi-cuadrado, Cramér's V

### 4.2.3 Análisis Multivariado

- **PCA:** Reducción de dimensionalidad lineal
- **ICA:** Análisis de componentes independientes
- **Clustering:** Detección de subgrupos

## 4.3 Entrenamiento de Modelos

La página  **Model Training** permite entrenar múltiples algoritmos.

### 4.3.1 Algoritmos Disponibles

| Categoría         | Algoritmo           | Características                   |
|-------------------|---------------------|-----------------------------------|
| Lineales          | Logistic Regression | Interpretable, baseline           |
|                   | SGD Classifier      | Escalable a big data              |
|                   | SVC (lineal)        | Margen máximo                     |
| Árboles           | Decision Tree       | Interpretable, propenso a overfit |
|                   | Random Forest       | Ensemble, robusto                 |
| Gradient Boosting | XGBoost             | Alto rendimiento, rápido          |
|                   | LightGBM            | Eficiente en memoria              |
|                   | CatBoost            | Maneja categóricas nativamente    |
| Instancia         | KNN                 | No paramétrico                    |
| Redes             | MLP                 | Captura no-linealidades           |

Cuadro 2: Algoritmos de clasificación disponibles

### 4.3.2 Validación Cruzada Rigurosa

El sistema implementa un pipeline de experimentación academicamente riguroso:

#### FASE 1 - Train + Validation:

- Repeated Stratified K-Fold ( $\geq 30$  runs)
- Estimación de  $\mu$  (media) y  $\sigma$  (desviación estándar)

- Curvas de aprendizaje

### FASE 2 - Test:

- Bootstrap resampling
- Jackknife (leave-one-out)
- Intervalos de confianza

### FASE 3 - Comparación Estadística:

- Test de normalidad (Shapiro-Wilk)
- Paired t-test o Mann-Whitney U
- Corrección de comparaciones múltiples

## 4.4 Evaluación de Modelos

La página [🔗 Model Evaluation](#) proporciona métricas exhaustivas.

### 4.4.1 Métricas de Clasificación

| Métrica               | Descripción                              | Rango                  |
|-----------------------|--|------------------------|
| AUROC                 | Área bajo curva ROC                      | [0, 1]                 |
| AUPRC                 | Área bajo curva Precision-Recall         | [0, 1]                 |
| Accuracy              | Proporción de aciertos                   | [0, 1]                 |
| Precision             | $VP / (VP + FP)$                         | [0, 1]                 |
| Recall (Sensibilidad) | $VP / (VP + FN)$                         | [0, 1]                 |
| Specificity           | $VN / (VN + FP)$                         | [0, 1]                 |
| F1-Score              | Media armónica Precision-Recall          | [0, 1]                 |
| Brier Score           | Error cuadrático medio de probabilidades | [0, 1] (menor = mejor) |

Cuadro 3: Métricas de evaluación disponibles

### 4.4.2 Curvas de Calibración

Las curvas de calibración evalúan si las probabilidades predichas corresponden con las frecuencias observadas:

- **Modelo bien calibrado:** Curva cercana a la diagonal

- **Sobreconfianza:** Curva por debajo de la diagonal
- **Subconfianza:** Curva por encima de la diagonal

#### 4.4.3 Decision Curve Analysis (DCA)

El DCA evalúa la utilidad clínica comparando con estrategias extremas:

- **Treat All:** Tratar a todos los pacientes
- **Treat None:** No tratar a nadie
- **Modelo:** El beneficio neto del modelo

### 4.5 Explicabilidad Avanzada (SHAP)

La página [🔍 Explainability](#) implementa análisis SHAP completo.

#### 4.5.1 Visualizaciones SHAP

1. **Beeswarm Plot:** Distribución de valores SHAP por feature
2. **Bar Plot:** Importancia media absoluta
3. **Waterfall Plot:** Contribución de cada feature para una predicción
4. **Force Plot:** Visualización interactiva de predicciones individuales

#### 4.5.2 Interpretación de SHAP

- **SHAP value positivo:** Aumenta la probabilidad de la clase positiva
- **SHAP value negativo:** Disminuye la probabilidad
- **Magnitud:** Indica la fuerza del efecto

### 4.6 AutoML

La página [🤖 AutoML](#) automatiza la búsqueda de modelos.

#### 4.6.1 Backends Disponibles

| Backend         | Plataforma     | Características                    |
|-----------------|----------------|------------------------------------|
| FLAML           | Cross-platform | Rápido, ligero, 12+ estimadores    |
| Auto-sklearn    | Linux/WSL      | Meta-learning, ensemble automático |
| AutoKeras (NAS) | Cross-platform | Neural Architecture Search         |

#### 4.6.2 Presets de Búsqueda

- **Quick** (5 min): Exploración rápida para prototipado
- **Balanced** (1 hora): Balance entre tiempo y rendimiento
- **High Performance** (4 horas): Búsqueda exhaustiva

### 4.7 Optimización Inversa

La página [🔗 Inverse Optimization](#) permite análisis “what-if”.

#### 4.7.1 Concepto

La optimización inversa responde preguntas como:

*“¿Qué valores de presión arterial y medicación minimizarían el riesgo de mortalidad para este paciente?”*

#### 4.7.2 Configuración

1. **Variables fijas:** Características no modificables (edad, sexo)
2. **Variables modificables:** Parámetros que pueden optimizarse (PA, medicación)
3. **Restricciones:** Límites clínicamente plausibles
4. **Objetivo:** Probabilidad target de mortalidad

## 5 Guía para Desarrolladores

### Sección para Desarrolladores

Esta sección requiere conocimientos de Python y desarrollo de software.

## 5.1 Modelos Personalizados

La página [🔧 Custom Models](#) permite crear arquitecturas propias.

### 5.1.1 Arquitectura Base

Todos los modelos personalizados deben heredar de las clases base:

```
from src.models.custom_base import BaseCustomClassifier

class MiModelo(BaseCustomClassifier):
    def __init__(self, param1=10, name="MiModelo"):
        super().__init__(name=name)
        self.param1 = param1

    def fit(self, X, y):
        self._validate_input(X, training=True)
        # ... lógica de entrenamiento ...
        self.is_fitted_ = True
        return self

    def predict(self, X):
        self._validate_input(X)
        # ... lógica de predicción ...
        return predictions

    def predict_proba(self, X):
        # ... probabilidades ...
        return probabilities
```

### 5.1.2 Métodos Requeridos

| Método                        | Tipo           | Descripción                     |
|-------------------------------|----------------|---------------------------------|
| <code>__init__()</code>       | Obligatorio    | Constructor con hiperparámetros |
| <code>fit(X, y)</code>        | Obligatorio    | Entrenar el modelo              |
| <code>predict(X)</code>       | Obligatorio    | Predicciones de clase           |
| <code>predict_proba(X)</code> | Clasificadores | Probabilidades por clase        |
| <code>get_params()</code>     | Recomendado    | Obtener hiperparámetros         |
| <code>set_params()</code>     | Recomendado    | Establecer hiperparámetros      |



### 5.1.3 Flujo de Trabajo

1. Crear el código Python del modelo
2. Guardarlo en `src/models/custom/`
3. Ir a [02 !\[\]\(38441ceaa711016e0bf2ad46ad394ff4\_img.jpg\) Model Training](#)
4. Activar “Include Custom Models” en la barra lateral
5. Seleccionar el modelo y entrenar

## 5.2 Despliegue con Docker

### 5.2.1 Construcción de la Imagen

```
cd Tools/docker
docker build -t mortality-ami-predictor ..
```

### 5.2.2 Ejecución del Contenedor

```
docker run -p 8501:8501 \
  -v $(pwd)/DATA:/app/DATA:ro \
  -v $(pwd)/models:/app/models \
  mortality-ami-predictor
```

### 5.2.3 Docker Compose

Para entorno de desarrollo completo:

```
docker-compose --profile dev up -d
```

Esto levanta:

- **app** (puerto 8501): Dashboard Streamlit
- **jupyter** (puerto 8888): Jupyter Lab para desarrollo
- **mlflow** (puerto 5000): Tracking de experimentos

### 5.3 Estructura del Código

```
Tools/
+-- dashboard/
|   +-- Dashboard.py           # Entrada principal
|   +-- app/                   # Utilidades y estado
|   +-- pages/                 # Páginas del dashboard
+-- src/
|   +-- data_load/             # Carga de datos
|   +-- cleaning/              # Limpieza
|   +-- eda/                   # Análisis exploratorio
|   +-- features/              # Ingeniería de features
|   +-- models/                # Definiciones de modelos
|   +-- training/              # Pipeline de entrenamiento
|   +-- evaluation/            # Métricas y evaluación
|   +-- explainability/        # SHAP y explicabilidad
|   +-- automl/                # AutoML integrations
|   +-- scoring/               # Scores clínicos
+-- tests/                     # Tests automatizados
+-- docker/                    # Configuración Docker
```

### 5.4 API Programática

Para integración con sistemas externos:

```
# Carga de datos
from src.data_load import load_dataset, train_test_split

# Limpieza
from src.cleaning import DataCleaner, CleaningConfig




# Entrenamiento
from src.training import run_rigorous_experiment_pipeline

# Predicción
from src.models import make_classifiers
model = make_classifiers()['xgboost']
model.fit(X_train, y_train)
predictions = model.predict_proba(X_test)

# Explicabilidad
```

```
from src.explainability import compute_shap_values
shap_values = compute_shap_values(model, X_test)
```

## 6 Referencia de Páginas del Dashboard

| #  | Página               | Funcionalidad                             |  |  |  |
|----|----------------------|---|---|---|---|
| 00 | Data Cleaning & EDA  | Limpieza de datos y análisis exploratorio | ✓   | ✓   | ✓   |
| 01 | Data Overview        | Resumen y estadísticas del dataset        | ✓   |   |   |
| 02 | Model Training       | Entrenamiento con validación cruzada      | ✓   | ✓   | ✓   |
| 03 | Predictions          | Predicciones individuales y por lotes     | ✓   | ✓   |   |
| 04 | Model Evaluation     | Métricas, ROC, calibración, DCA           |   | ✓   |   |
| 05 | Explainability       | SHAP, permutación, PDP                    | ✓   | ✓   |   |
| 06 | Clinical Scores      | GRACE y TIMI                              | ✓   |   |   |
| 07 | Custom Models        | Creación de modelos personalizados        |   |   | ✓   |
| 08 | Inverse Optimization | Optimización de features                  |   | ✓   | ✓   |
| 09 | AutoML               | Entrenamiento automático                  |   | ✓   |   |

Cuadro 4: Páginas del dashboard y perfiles de usuario recomendados

**Leyenda:**  Personal Clínico |  Científicos de Datos |  Desarrolladores

## 7 Solución de Problemas

### 7.1 Problemas Comunes

| Problema                            | Solución  |
|-------------------------------------|---|
| “No hay datos cargados”             | Ir a la página 00 y cargar un dataset primero                                       |
| “No trained models found”           | Entrenar modelos en la página 02 antes de usar predicciones                         |
| Error de codificación al cargar CSV | El sistema intenta múltiples encodings automáticamente. Si falla, convertir a UTF-8 |
| SHAP tarda mucho                    | Reducir el número de muestras en la barra lateral                                   |
| AutoML no disponible                | Instalar FLAML: <code>pip install flaml[automl]</code>                              |
| Auto-sklearn no funciona            | Solo disponible en Linux. Usar FLAML en Windows/macOS                               |
| Memoria insuficiente                | Reducir tamaño del dataset o usar submuestreo                                       |

## 7.2 Mensajes de Error Frecuentes

### ⚠ Target column not found

**Causa:** El nombre de la columna objetivo no coincide con la configuración.

**Solución:** Verificar que el dataset contiene una columna llamada `mortality_inhospital` o configurar la variable de entorno `TARGET_COLUMN`.

### ⚠ Module not found: shap

**Causa:** La librería SHAP no está instalada.

**Solución:** Ejecutar `pip install shap` en el entorno virtual.

## 8 Glosario

**AUROC** Área Bajo la Curva ROC (Receiver Operating Characteristic). Mide la capacidad discriminativa del modelo.

**AUPRC** Área Bajo la Curva Precision-Recall. Útil para datasets desbalanceados.

**AutoML** Machine Learning Automatizado. Búsqueda automática de algoritmos e hiperparámetros.

### Calibración

Grado en que las probabilidades predichas coinciden con las frecuencias observadas.

**DCA** Decision Curve Analysis. Evalúa la utilidad clínica de un modelo.

**Feature** Variable o característica utilizada como entrada del modelo.

**FLAML** Fast and Lightweight AutoML. Framework de AutoML de Microsoft.

**GRACE** Global Registry of Acute Coronary Events. Score clínico para SCA.

**IAM** Infarto Agudo de Miocardio.

**ICA** Independent Component Analysis. Técnica de reducción de dimensionalidad.

**Killip** Clasificación de insuficiencia cardíaca en IAM (I-IV).

**NAS** Neural Architecture Search. Búsqueda automática de arquitecturas de redes neuronales.

**PCA** Principal Component Analysis. Reducción de dimensionalidad lineal.

**SHAP** SHapley Additive exPlanations. Método de explicabilidad basado en teoría de juegos.

**TIMI** Thrombolysis In Myocardial Infarction. Score de riesgo cardiovascular.

## 9 Contacto y Soporte

### 🔗 ¿Necesitas ayuda?

🔗 <https://github.com/Pol4720/mortality-ami-predictor>

📖 Documentación técnica en `Tools/docs/`

🐛 Reportar issues en GitHub

---

Mortality AMI Predictor v2.0

Desarrollado con ❤️ para la investigación médica

Diciembre 2025