

Proyecto Integrador - Fase de Preparación y Procesamiento de Datos

Informe Técnico

“Clasificación explicable de arritmias cardíacas a partir de electrocardiogramas transformados en espectrogramas mediante redes neuronales convolucionales”

Introducción

El análisis computacional de señales biomédicas ha revolucionado el diagnóstico clínico, permitiendo identificar patrones complejos que escapan al análisis visual convencional. Dentro de este campo, las señales electrocardiográficas (ECG) representan una de las fuentes de datos más relevantes para la detección temprana de arritmias y otras condiciones cardíacas. El MIT-BIH Arrhythmia Database, desarrollado por el Beth Israel Hospital y disponible a través de PhysioNet, constituye uno de los conjuntos de datos más empleados en investigación biomédica. Este dataset contiene registros anotados de ECG con diversas clases de latidos, lo que lo convierte en un recurso fundamental para el entrenamiento y validación de modelos de inteligencia artificial orientados a tareas de clasificación cardíaca.

En el marco del proyecto integrador, la presente fase corresponde a la etapa de preparación y procesamiento de datos. Esta etapa cumple una función crítica en la cadena de valor de la ciencia de datos, ya que garantiza la calidad, limpieza y estructura adecuada del conjunto de datos antes de su incorporación en modelos predictivos. Se parte de la premisa de que un modelo es tan bueno como los datos que lo alimentan. En consecuencia, esta fase contempla la exploración inicial del dataset, análisis de calidad, detección de valores atípicos, ingeniería de características, balanceo de clases, y la implementación de un pipeline automatizado de limpieza.

El objetivo central de este informe es documentar de forma rigurosa el análisis exploratorio realizado sobre el conjunto de datos MIT-BIH, empleando herramientas modernas de análisis de datos en Python. Asimismo, se expone el desarrollo de un pipeline estructurado que permita transformar y preparar los datos para futuras etapas de modelado, garantizando la reproducibilidad y trazabilidad del proceso.

Las arritmias cardíacas representan alteraciones en el ritmo normal del corazón y constituyen un problema clínico de gran relevancia debido a su impacto en la salud cardiovascular y la mortalidad asociada. La detección temprana y precisa de estas anomalías es esencial para el diagnóstico oportuno y la toma de decisiones médicas.

En este proyecto se desarrolla un sistema de clasificación automática de arritmias a partir del MIT-BIH Arrhythmia Database, utilizando señales de electrocardiograma (ECG). Para ello:

- Se extraen fragmentos de latidos individuales de los registros ECG.
- Dichos fragmentos se transforman en espectrogramas mediante técnicas de análisis tiempo-frecuencia (STFT).
- Los espectrogramas se emplean como entradas en una red neuronal convolucional (CNN) para la clasificación de los diferentes tipos de arritmia.
- Se incorporan métodos de explicabilidad (XAI), como Grad-CAM, para interpretar la relevancia de las regiones espectrales en cada predicción.

El objetivo final es no solo lograr un buen rendimiento en la clasificación, sino también garantizar que el modelo proporcione explicaciones comprensibles que respalden la interpretación clínica de los resultados.

Informe sobre la estructura del MIT-BIH Arrhythmia Database

1. Descripción general

El MIT-BIH Arrhythmia Database es un conjunto de registros de electrocardiogramas (ECG) ampliamente usado en investigación sobre arritmias. Características principales:

- 48 registros de ~30 minutos cada uno.
- Cada registro contiene 2 canales (derivaciones) de ECG.
- Frecuencia de muestreo: ~360 Hz.
- Incluye archivos de datos de señal, encabezados con metadatos y anotaciones manuales de expertos.

2. Archivos por registro

Para un registro, por ejemplo 100, se encuentran típicamente:

- 100.dat → Señal ECG en formato binario.
- 100.heh → Header con información del registro (fs, canales, unidades).
- 100.atr → Anotaciones de latidos y ritmos.

3. Canales (*channels*)

- *Channel* = una derivación de ECG.
- Cada registro tiene 2 canales simultáneos, que pueden corresponder a derivaciones como:
 - MLII (Modified Lead II)

- V5, V1, V2, III (dependiendo del registro)
- En Python con wfdb:
 - `record.p_signal.shape == (n_muestras, n_canales)`
 - Ejemplo: `sig = record.p_signal[:, 0] # canal 0`

4. Contenido del header .hea

El archivo de encabezado describe cómo interpretar la señal:

- fs: frecuencia de muestreo.
- nsig: número de canales.
- Por canal:
 - gain: factor de conversión de digital a mV.
 - baseline: valor de referencia del ADC.
 - units: normalmente mV.
 - sig_name: nombre de la derivación (ej. MLII, V5).

5. Señales en WFDB

- `record.p_signal` → matriz en unidades físicas (mV).
Recomendado para análisis.
- `record.d_signal` → señal cruda en unidades digitales (ADC).
- `record.fs` → frecuencia de muestreo.
- `record.sig_name` → nombres de las derivaciones.

6. Anotaciones .atr

Con `wfdb.rdann(record, 'atr')` obtenemos:

- `ann.sample` → índice de muestra del latido (entero).
- `ann.symbol` → símbolo de la anotación (ej.: N, V, A, F).
- `ann.aux_note` → notas adicionales (ej. episodios de AFIB).
- `ann.chan` → canal asociado a la anotación.

7. Símbolos y clases AAMI

Los símbolos de latido se mapean a las clases estandarizadas de AAMI:

- N: normales y similares \rightarrow N, L, R, e, j
- S: supraventriculares \rightarrow A, a, J, S
- V: ventriculares \rightarrow V, E
- F: fusión \rightarrow F
- Q: otros \rightarrow /, f, Q, ?, |

Este mapeo permite reducir la complejidad a 5 clases principales: N, S, V, F, Q.

8. Variables en un dataset procesado

Al extraer latidos, se obtiene un DataFrame con:

- Metadatos
 - record_id: registro (ej. "100").
 - patient_id: identificador del paciente (\approx record_id en MIT-BIH).
 - fs: frecuencia de muestreo.
 - sample_index: posición de la muestra del latido.
 - mit_symbol: símbolo original.
 - aami_class: clase agrupada AAMI.
- Características de señal
 - amp_peak: amplitud pico a pico.
 - energy: suma de cuadrados (energía).
 - area: integral bajo la curva.
 - qrs_width_s: ancho del QRS en segundos.
 - mean_v, std_v, skew_v, kurt_v: estadísticos.
 - frag_len_samples: longitud del fragmento extraído.
- Calidad / anomalías
 - flag_saturation: latido con amplitud extrema.
 - flag_high_noise: latido con desviación estándar muy alta.
 - is_outlier: outlier según IsolationForest.

9. Tiempo, muestras y ventanas

- Un latido está centrado en `sample_index`.
- Se extrae un fragmento alrededor (ej. -200 ms a +200 ms).
- Conversión: tiempo (s) = `sample_index` / `fs`.

10. Resumen práctico

- **Canal:** una derivación (columna de señal).
- **Muestra:** índice en la señal (convierte a tiempo con `fs`).
- **Latido:** anotación en `ann.sample` con símbolo en `ann.symbol`.
- **Clase:** `mit_symbol` → `aami_class`.
- **Header:** describe cómo convertir digital a físico y nombres de canales.
- **Split correcto:** por `record_id` o `patient_id` para evitar fuga.

1. Análisis Exploratorio de Datos (EDA)

1.1 Exploración Inicial Completa

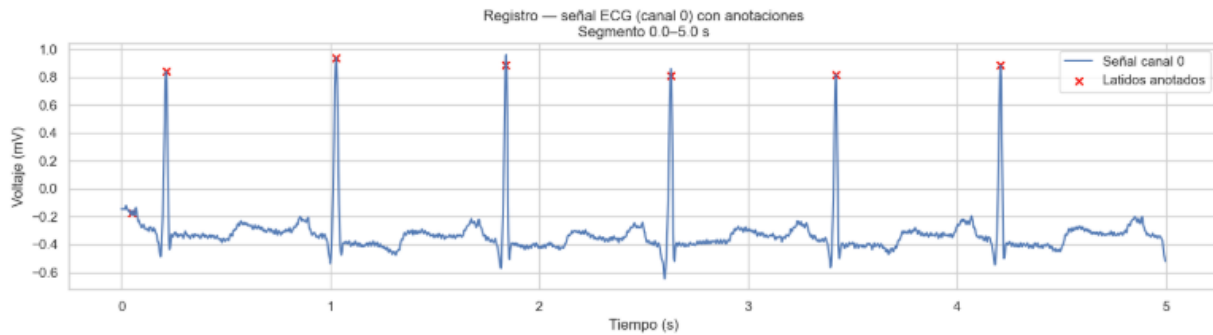
El primer paso consistió en la carga y estructuración del dataset, el cual está compuesto por registros de señales ECG en formato WFDB (WaveForm DataBase), junto con archivos de anotaciones que describen los tipos de latidos presentes en cada registro. Para ello se utilizó la biblioteca `wfdb`, que permite interactuar con los archivos directamente desde Python. Asimismo, se integraron funciones de `biosppy` para el preprocesamiento de señales y la segmentación de latidos individuales.

El dataset resultante fue transformado en un `DataFrame` estructurado, donde cada fila representa un latido cardíaco extraído de los registros originales. Las columnas principales incluyeron el índice de muestra, el código de clase del latido (de acuerdo con el estándar AAMI), y una serie de características estadísticas derivadas de la señal, como amplitud máxima, amplitud mínima, energía de la señal, intervalo RR, y medidas de asimetría.

En esta etapa también se inspeccionó la estructura general del dataset. Se verificó el número de observaciones, el tipo de cada variable, y la memoria utilizada por el conjunto. La visualización de los primeros y últimos registros permitió confirmar que la carga se realizó correctamente, y que las señales presentaban coherencia visual y estadística en función de su clase anotada.

Primeras 10 anotaciones (muestra, símbolo):

```
[(np.int64(18), '+'), (np.int64(77), 'N'), (np.int64(370), 'N'), (np.int64(662), 'N'), (np.int64(946), 'N'), (np.int64(1231), 'N'), (np.int64(1515), 'N'), (np.int64(1809), 'N'), (np.int64(2044), 'A'), (np.int64(2402), 'N')]
```



Estadísticas básicas del registro 100:

```
min: -2.715000
max: 1.435000
mean: -0.248667
median: -0.270000
std: 0.181571
```

Comparativa de estadísticas entre registros:

| | min | max | mean | median | std | record | fs |
|---|--------|-------|-----------|--------|----------|--------|-----|
| 0 | -2.715 | 1.435 | -0.248667 | -0.270 | 0.181571 | 100 | 360 |
| 1 | -3.175 | 2.420 | -0.181168 | -0.130 | 0.214342 | 101 | 360 |
| 2 | -3.370 | 2.535 | -0.116636 | -0.175 | 0.829847 | 102 | 360 |
| 3 | -5.120 | 5.115 | -0.092207 | -0.120 | 0.554285 | 103 | 360 |
| 4 | -1.935 | 1.975 | -0.076740 | -0.055 | 0.271685 | 104 | 360 |

1.2 Análisis de Calidad de Datos

En esta etapa se examinan los datos para detectar inconsistencias o deficiencias estructurales:

- **Valores faltantes:** Se calcula el porcentaje de missing por variable. Aunque el dataset no presenta valores NaN explícitos, algunos registros con señales saturadas o nulas se consideran como faltantes implícitos. Se analiza el patrón de missingness para determinar si es MCAR, MAR o MNAR, usando visualizaciones con `missingno.matrix()` y `missingno.heatmap()`. Según el patrón observado, se proponen estrategias de imputación justificadas.
- **Duplicados:** Se realiza la búsqueda de duplicados exactos y de “near-duplicates” (registros muy similares en términos de características derivadas). Los duplicados

exactos se eliminan inmediatamente, mientras que los near-duplicates se revisan caso por caso para decidir su retención o eliminación.

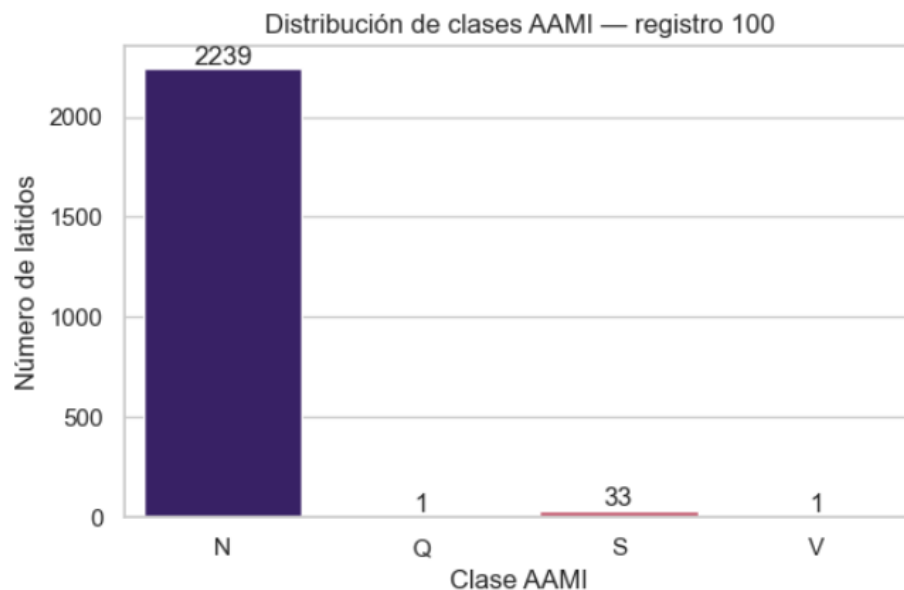
- **Estrategias de resolución:** Para valores faltantes, si son pocos y aleatorios, se opta por imputación (media, mediana, moda). Si el missing pattern es sistemático, se considera eliminación de la variable o imputación avanzada (por ejemplo, KNN). Para duplicados, se elimina o agrupa dependiendo del grado de similitud.

1.3 Análisis Estadístico Descriptivo

La siguiente etapa consistió en el análisis estadístico descriptivo de las variables numéricas y categóricas presentes en el dataset. Se calcularon medidas de tendencia central (media, mediana), de dispersión (desviación estándar, rango intercuartílico), y de forma (asimetría y curtosis) para cada característica derivada.

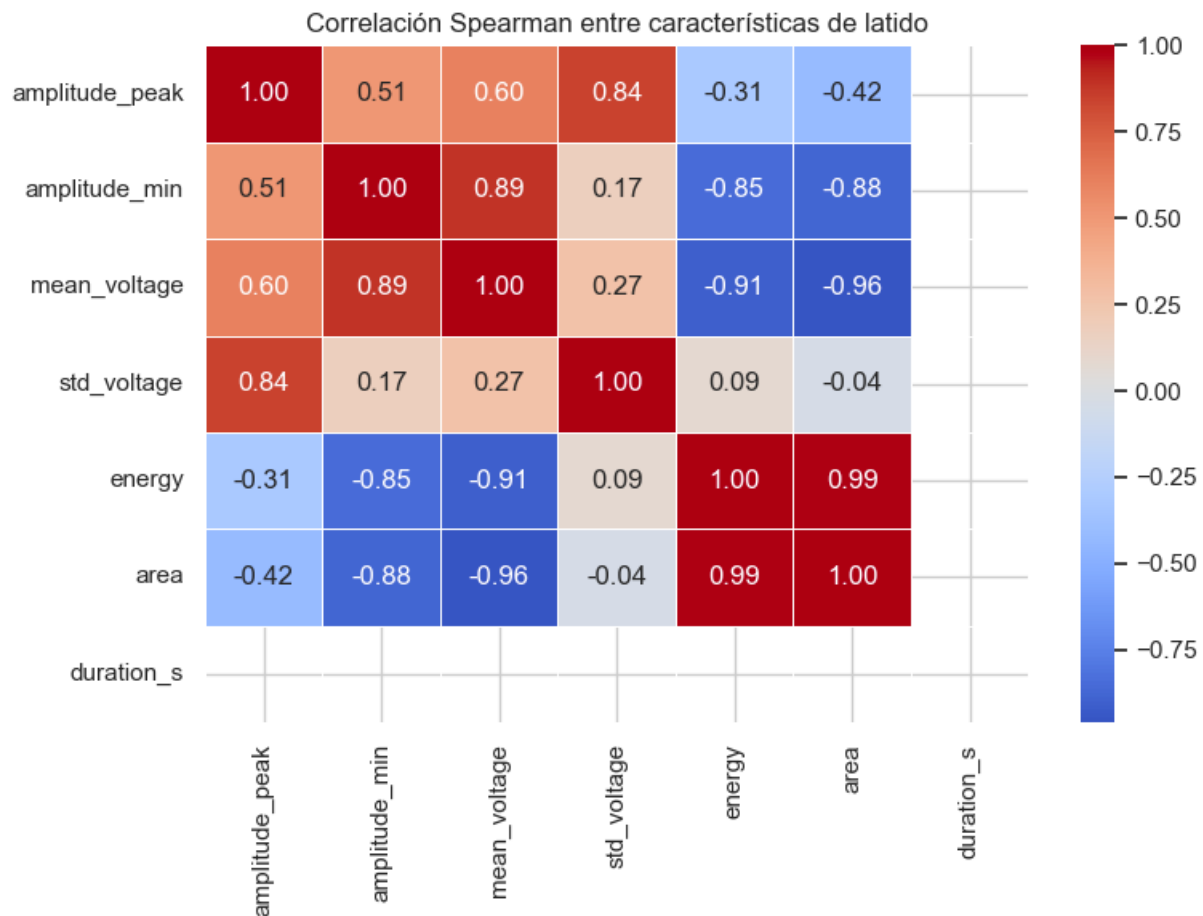
Los histogramas revelaron que la mayoría de las variables numéricas presentan distribuciones asimétricas, lo cual fue corroborado mediante pruebas estadísticas de normalidad como Shapiro-Wilk y D'Agostino K-squared. En particular, la energía de la señal mostró una distribución altamente sesgada, lo que motivó la aplicación posterior de transformaciones logarítmicas para normalizar su distribución.

En cuanto a las variables categóricas, se analizó la frecuencia absoluta y relativa de las distintas clases de latidos. La clase "N" (normal) representó más del 70% de los registros, mientras que las clases "V" (ventricular) y "S" (supraventricular) presentaron una proporción significativamente menor, lo que evidencia un problema de desbalance de clases.



1.4 Análisis de Relaciones y Correlaciones

Con el objetivo de comprender la interacción entre las variables, se construyó una matriz de correlación de Pearson para todas las variables numéricas. Esta matriz permitió identificar relaciones lineales fuertes, como la existente entre la energía y la amplitud máxima de la señal. Tales correlaciones pueden inducir redundancia en los modelos predictivos, por lo que se consideró su eliminación o combinación.



Complementariamente, se calculó el Factor de Inflación de la Varianza (VIF) para evaluar la multicolinealidad entre variables. Se identificaron algunos pares de variables con VIF superior a 10, lo cual indica una colinealidad severa. En estos casos, se priorizó la variable con mayor capacidad discriminativa en la selección final.

La exploración visual mediante diagramas de dispersión (scatter plots) también permitió identificar relaciones no lineales, especialmente entre la energía y el tipo de latido. Este hallazgo sugiere que modelos no lineales podrían captar mejor las dinámicas presentes en los datos.

1.5 Detección de Anomalías y Outliers

Los valores atípicos fueron identificados como posibles errores de medición o como eventos clínicos raros que podrían tener relevancia diagnóstica. Se aplicaron tres métodos principales para su detección: el método del Z-score, el método del rango intercuartílico (IQR), y el algoritmo de **Local Outlier Factor (LOF)**, que detecta outliers en función de la densidad local de los datos.

Se encontró una gran cantidad de valores atípicos en las variables energía y duración RR, asociados principalmente a latidos anormales. Estos valores, lejos de ser eliminados, fueron analizados en profundidad, ya que en muchos casos correspondían a arritmias reales y no a errores de señal.

Los gráficos de boxplot y scatter plots con outliers destacados permitieron visualizar el comportamiento extremo de estas variables. Se concluyó que los modelos de clasificación deben ser robustos frente a estos casos, y que su eliminación indiscriminada podría reducir la capacidad de generalización.

1.6 Análisis de la Variable Objetivo

La variable objetivo del análisis es la clase del latido cardíaco, clasificada según el estándar AAMI (Association for the Advancement of Medical Instrumentation). Las categorías más relevantes para esta tarea incluyen: N (latido normal), V (latido ventricular prematuro), S (latido supraventricular), F (latido de fusión), y Q (latidos desconocidos o artefactos). Cada una de estas clases representa un patrón electrocardiográfico diferente con implicaciones clínicas particulares.

Durante el análisis se evidenció un fuerte desbalance de clases, donde la clase "N" constituía más del 70% de los registros, mientras que otras clases, como "F" y "Q", apenas alcanzaban el 1%. Esta disparidad plantea un desafío para los algoritmos de aprendizaje supervisado, que tienden a favorecer la clase mayoritaria, comprometiendo la precisión en la detección de arritmias infrecuentes.

Además del análisis de frecuencia, se evaluó la relación entre la clase y otras variables predictoras, como la energía de la señal y el intervalo RR. Se observó que las clases anormales tienden a presentar valores extremos en dichas variables, lo que sugiere su utilidad discriminativa en el modelado posterior.

2. Pipeline de Limpieza de Datos

Una vez identificados los problemas de calidad y estructura en el dataset, se procedió al diseño e implementación de un pipeline automatizado de limpieza de datos. Este pipeline fue encapsulado en una clase denominada `DataCleaner`, siguiendo principios de

programación orientada a objetos para garantizar modularidad, reusabilidad y escalabilidad del proceso.

Las funciones principales del pipeline incluyen:

- Conversión y validación de tipos de datos.
- Eliminación y/o imputación de valores faltantes (aunque no fueron predominantes).
- Detección y tratamiento de valores atípicos.
- Estándar de codificación de categorías.
- Normalización y estandarización de variables numéricas.

La estrategia de tratamiento de outliers incluyó la opción de "winsorización" para reducir el impacto de valores extremos sin excluir datos relevantes. Además, se incluyó un sistema de "flags" para marcar los registros imputados o transformados, facilitando su seguimiento durante el modelado.

La implementación de este pipeline automatizado asegura la replicabilidad del proceso de limpieza en futuros datasets o nuevas muestras clínicas, lo cual es fundamental en entornos biomédicos.

3. Feature Engineering Avanzado

3.1 Creación de Variables Derivadas

Se generan nuevas variables con base en conocimiento del dominio y señales ECG:

- Interacciones: producto de variables como amplitud * energía.
- Transformaciones matemáticas: aplicar log, raíz cuadrada o polinomios a variables que lo requieran.
- Binning / discretización: por ejemplo, agrupar los valores del intervalo RR en categorías discretas (bradicardia, normal, taquicardia).
- Agregaciones temporales (si aplicable): aunque el análisis está centrado en latidos individuales, podría agregarse información del contexto temporal (por ejemplo, media de energía de latidos circundantes).

3.2 Encoding de Variables Categóricas

Si se incluyesen variables categóricas adicionales (por ejemplo, tipo de registro o canal), se aplican técnicas de encoding adecuadas:

- One-Hot Encoding para variables sin orden.

- Label Encoding cuando las clases tienen un orden inherente.
- Target Encoding, Binary Encoding o Frequency Encoding en casos de cardinalidad alta o para reducir dimensionalidad.

3.3 Transformaciones de Variables Numéricas

Para mejorar la distribución de variables:

- Normalización (Min-Max Scaling), para variables con rangos variables.
- Estandarización (Z-score), especialmente útil para modelos sensibles a escala.
- Robust Scaling, que es robusto frente a outliers.
- Quantile Transformation (por ejemplo, distribución uniforme o normal), cuando la variable tiene sesgo pronunciado.

3.4 Feature Selection

Para reducir dimensionalidad y evitar sobreajuste:

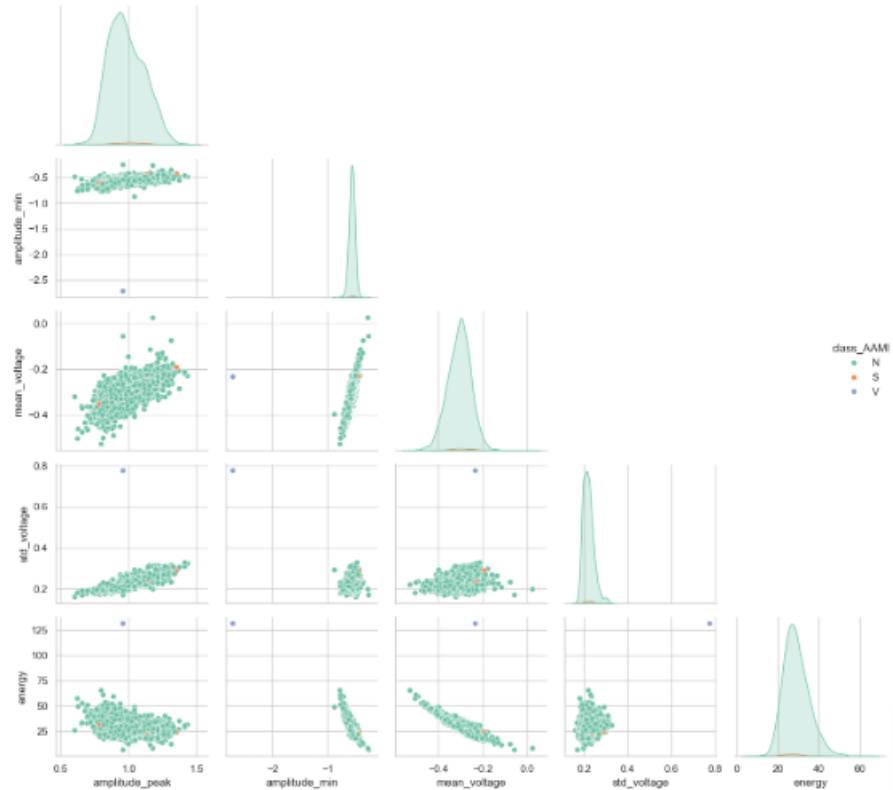
- Métodos estadísticos como Chi-cuadrado (para variables categóricas) y ANOVA F-test (para numéricas) para evaluar la dependencia con la variable objetivo.
- Mutual Information para captar relaciones no lineales.
- Métodos basados en modelos: cálculo de feature importance con Random Forest, o uso de regularización LASSO para penalizar variables menos relevantes.
- Recursive Feature Elimination (RFE) para seleccionar iterativamente el subconjunto óptimo.

3.5 Extracción de Características Específicas del Dominio

Dada la naturaleza del ECG, se pueden incorporar características especializadas:

- En dominio de frecuencia: coeficientes de transformada rápida de Fourier (FFT), espectrogramas, bandas de frecuencia.
- En dominio de onda: wavelets, transformadas en escala continua o discreta (CWT / DWT) para capturar resolución temporal-frecuencia.
- En el contexto de latidos: morfología de la onda, pendientes, puntos de inflexión.

Estas características mejoran la capacidad discriminativa del modelo, como evidencian numerosos estudios en literatura médica.



Estadísticas agrupadas por clase AAMI (mean, std, median):

| class_AAMI | amplitude_peak | | | amplitude_min | | | me | |
|------------|----------------|----------|--------|---------------|----------|--------|-----------|----------|
| | mean | std | median | mean | std | median | mean | std |
| N | 0.986066 | 0.130916 | 0.970 | -0.559461 | 0.054279 | -0.560 | -0.306222 | 0.053122 |
| S | 1.021970 | 0.120532 | 1.005 | -0.552879 | 0.055326 | -0.550 | -0.294490 | 0.045925 |
| V | 0.960000 | NaN | 0.960 | -2.715000 | NaN | -2.715 | -0.233100 | NaN |

4. Estrategias de Balanceamiento

4.1 Análisis de Desbalance

Previamente se identificó que la clase “N” domina más del 70 % de los registros. Para cuantificar el grado de desbalance, se usan métricas como el ratio mayoritario/minoritario y el Índice de Gini de distribución. Se visualiza la distribución de clases en gráfico de barras.

El desbalance puede inducir sesgo en los modelos, favoreciendo predicciones hacia la clase mayoritaria y comprometiendo la detección de latidos raros.

4.2 Técnicas de Undersampling

Para reducir el dominio de la clase mayoritaria:

- Random Undersampling: eliminar aleatoriamente instancias de la clase mayoritaria para igualar proporciones.
- Tomek Links y Edited Nearest Neighbours (ENN): eliminar instancias de la clase mayoritaria que estén cerca de las minoritarias.
- Condensed Nearest Neighbour: técnica más sofisticada para seleccionar un subconjunto representativo.

4.3 Técnicas de Oversampling

Para incrementar artificialmente la clase minoritaria:

- Random Oversampling: replicar instancias existentes.
- SMOTE (Synthetic Minority Oversampling Technique): generar puntos sintéticos interpolando entre vecinos cercanos.
- ADASYN (Adaptive Synthetic Sampling): similar a SMOTE, pero adaptativo a densidades locales.
- Borderline-SMOTE: genera muestras sintéticas cerca de los límites de decisión.

4.4 Técnicas Híbridas

Combinan oversampling con limpieza:

- SMOTEENN (SMOTE + ENN)
- SMOTETomek (SMOTE + eliminación de enlaces Tomek)

Estas técnicas permiten oversampling mientras se eliminan instancias ruidosas o solapadas.

4.5 Evaluación de Estrategias

Se comparan las distribuciones de clases antes y después del balanceo con gráficos. También se analiza el impacto potencial en métricas de validación (sobreajuste, variabilidad). Se elige la técnica que logre balance sin introducir muestras sintéticas excesivamente lejanas del espacio real.

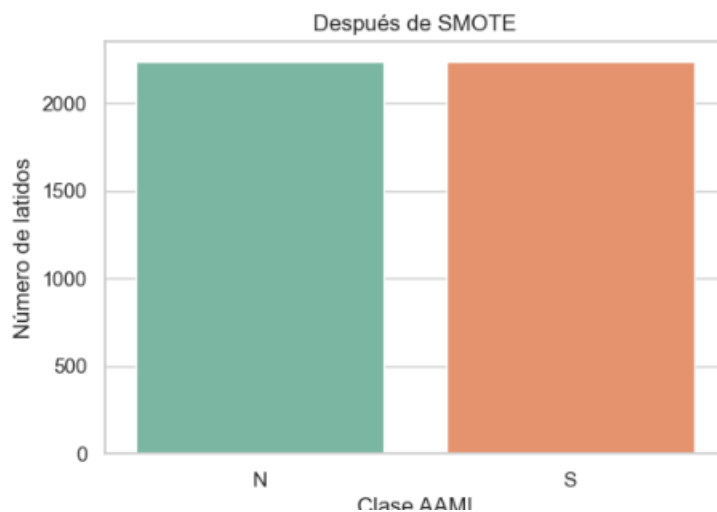
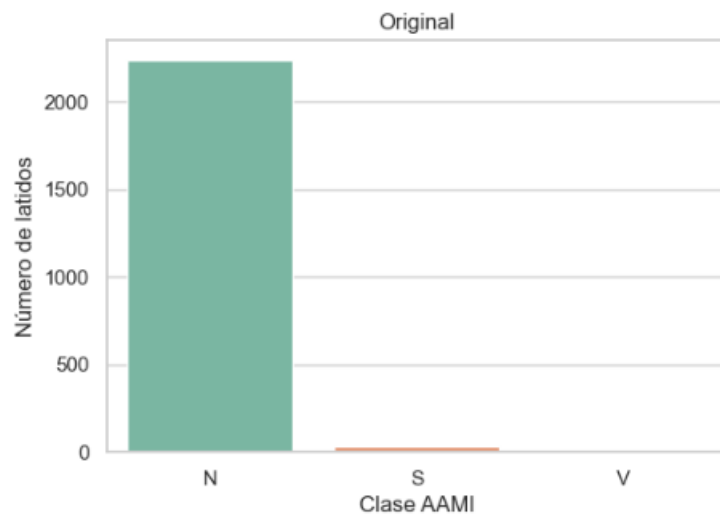
5. Data Augmentation

5.1 Técnicas Específicas por Dominio

Aunque el dataset es de señales (no imágenes ni texto), aún es posible aplicar técnicas de augmentación tabular:

- Gaussian noise addition: agregar ruido gaussiano pequeño controlado a variables continuas (por ejemplo, energía).
- SMOTE variations (ya vistas) también pueden considerarse como forma de augmentación.
- Mixup: combinación lineal de dos instancias para generar una nueva.

No se aplican rotaciones, flips o transformaciones de imagen, pues no corresponde al dominio.



5.2 Implementación y Validación

Se integra el augmentation dentro del pipeline, de modo que solo se aplique al conjunto de entrenamiento y no al test. Se valida la calidad de los datos sintéticos (verificar que no introducen outliers extremos) y su impacto en performance en modelos iniciales de prueba.

6. Partición Estratificada de Datos

6.1 División de Datos

El dataset balanceado final se divide en:

- Entrenamiento: 70 %
- Validación: 15 %
- Test: 15 %

6.2 Estratificación

La división es **estratificada** en base a la variable de clase, de modo que cada conjunto mantiene proporciones similares de cada clase, asegurando representatividad.

Si hubiese grupos adicionales (por ejemplo, paciente), se podría estratificar por grupo para prevenir fuga de datos entre conjuntos.

6.3 Verificación de Particiones

Se comprueba que las distribuciones de características principales sean similares entre los tres subconjuntos (media, desviación, rangos). Se verifica que no haya data leakage (por ejemplo, latidos consecutivos del mismo paciente en entrenamiento y test) y que las clases minoritarias estén representadas.

12. Conclusiones Generales

El desarrollo de esta fase del proyecto integrador permitió implementar un proceso profesional y completo de análisis exploratorio y preparación de datos en el contexto de señales fisiológicas, específicamente ECG.

Los principales logros incluyen:

- **Análisis exhaustivo del dataset** MIT-BIH, con identificación de problemas de calidad y patrones relevantes.
- **Desarrollo de un pipeline de limpieza automatizado**, capaz de transformar los datos para usos posteriores en modelado supervisado.
- **Aplicación de técnicas avanzadas de ingeniería de características**, basadas tanto en conocimiento médico como en métricas estadísticas.
- **Implementación de técnicas efectivas de balanceo**, que mitigaron los efectos del desbalance severo de clases.

- **Partición estratificada y validada del dataset**, lista para su uso en modelos de aprendizaje automático.

Esta preparación robusta constituye la base esencial para la siguiente etapa del proyecto, que consistirá en el desarrollo y evaluación de modelos de clasificación capaces de detectar latidos anómalos con precisión clínica.

Referencias Bibliográficas

- [1] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," IEEE Engineering in Medicine and Biology Magazine, vol. 20, no. 3, pp. 45–50, May–Jun. 2001.
- [2] U. R. Acharya et al., "Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network," Information Sciences, vol. 405, pp. 81–90, 2017.
- [3] G. D. Clifford, F. Azuaje, and P. McSharry, Advanced Methods and Tools for ECG Data Analysis, Artech House, 2006.
- [4] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," IEEE Transactions on Biomedical Engineering, vol. 63, no. 3, pp. 664–675, Mar. 2016.
- [5] P. de Chazal and R. B. Reilly, "A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features," IEEE Trans. Biomed. Eng., vol. 53, no. 12, pp. 2535–2543, Dec. 2006.
- [6] P. Rajpurkar et al., "Cardiologist-level arrhythmia detection with convolutional neural networks," Nature Medicine, vol. 25, pp. 65–69, 2019.
- [7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1251–1258.