

# ECG Heartbeat Categorization Dataset

Grace Aviance<sup>1</sup>[A01285158], Carlos Sánchez<sup>[A01640495]</sup>, Fabian Lioner<sup>1</sup>[A01633776],  
and Samuel Padilla Esqueda<sup>1</sup>[A01641383]

Tec de Monterrey, Av. Gral Ramón Corona No. 2514 Zapopan, Jal.  
<https://github.com/Elcasvi/ECG-Heartbeat-Categorizationj>

**Abstract.** ECG Heartbeat Categorization Dataset Analysis.

Este trabajo aborda el desarrollo de un modelo de clasificación para identificar diversas afecciones cardíacas a partir de lecturas de electrocardiogramas (ECG).

Dado que las enfermedades cardiovasculares son la principal causa de mortalidad en México, este proyecto se centra en mejorar la detección temprana de condiciones como arritmias, con el fin de reducir las tasas de mortalidad y mejorar la atención médica.

Se utilizan cuatro conjuntos de datos, compuestos por señales de ECG de pacientes sanos y con enfermedades cardíacas, para entrenar y probar varios algoritmos de clasificación supervisada. Se analizan problemas como la correlación entre los datos y el desequilibrio de clases, los cuales podrían afectar el rendimiento del modelo.

Entre los modelos evaluados, el algoritmo KNeighbors obtuvo los mejores resultados con una precisión de 96.97%. Este enfoque demuestra la efectividad de los modelos de aprendizaje automático para la clasificación de eventos cardíacos, ofreciendo una herramienta útil para el diagnóstico temprano de enfermedades del corazón.

**Keywords:** Machine Learning models · Classification.

## 1 Introducción

Las enfermedades cardiovasculares representan uno de los principales retos de salud pública en México, actualmente, las enfermedades del corazón ocupan los primeros lugares en cuanto a causas de mortalidad y según datos recientes, las enfermedades del corazón, como el infarto de miocardio, fueron responsables del 12.7% de las muertes en 2014.

A lo largo de los últimos 70 años, las enfermedades del corazón se han convertido en las principales causas de muerte en el país. Esto se debe a factores como: el envejecimiento poblacional, el incremento en el sobrepeso y la obesidad, el tabaquismo y la hipertensión arterial. Estos factores de riesgo afectan a un porcentaje preocupante de la población, ya que más del 70% de los adultos mexicanos presentan alguna de estas condiciones que incrementan el riesgo de sufrir enfermedades cardíacas.

Es por esto que el desarrollo de herramientas tecnológicas que puedan asistir en la detección y diagnóstico temprano de enfermedades del corazón es de gran

relevancia. Este proyecto se centra en el entrenamiento de un modelo capaz de identificar diferentes afecciones cardíacas, como arritmias, a partir de lecturas de electrocardiogramas (ECG). Con el objetivo de lograr una detección temprana de estas enfermedades, mejorar la atención médica y reducir las tasas de mortalidad.

## 2 ECG

Un ECG (electrocardiograma) es una prueba médica no invasiva que registra la actividad eléctrica del corazón a lo largo del tiempo, para diagnosticar diversas afecciones cardíacas.

El siguiente sistema de clasificación da una interpretación del ECG para categorizar diferentes tipos de latidos cardíacos o eventos cardíacos.

**N** significa ritmo sinusal normal, lo que indica un latido cardíaco regular y coordinado que se origina en el nódulo sinusal.

**S** representa arritmias supraventriculares, que son ritmos cardíacos anormales.

**Q** denota latidos cardíacos desconocidos / no clasificados / patrones que no se pueden categorizar con seguridad.

**V** indica arritmias ventriculares, que son ritmos cardíacos anormales.

**F** representa latidos de fusión, que presentan características tanto de ritmos cardíacos normales como anormales.

## 3 Dataset

### 3.1 Descripción del Dataset

Consiste en una serie de archivos csv con sus respectivas matrices de información con valores numéricos que tienen un rango de entre 0 y 1. Las columnas no indican nombre alguno que ayude a entender por ejemplo la segmentación temporal de cada registro. Además, la ultima columna denota cuál clase es ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4]. Los 4 datasets:

- mitbih\_train : (21892 filas, 188 columnas) Para el entrenamiento del modelo
- mitbih\_test : (87554, 188) Datos para probar la calidad del modelo
- ptbdb\_abnormal : (10506, 188) Una vez entrenado el modelo, lo probaremos en este dataset para corroborar su calidad con datos nuevos que indican enfermedades cardiacas
- ptbdb\_normal : (4046, 188) Contiene datos de pruebas sanas

### 3.2 Entendimiento de los datos

El Instituto Nacional de Metrología de Alemania (PTB - Physikalisch-Technische Bundesanstalt) realizó esta compilación de ECG digitalizados para fines de investigación, evaluación comparativa algorítmica o enseñanza a los usuarios de PhysioNet. Los ECG fueron recopilados de voluntarios sanos y pacientes con

diferentes enfermedades cardíacas por el profesor Michael Oeff, M.D., del Departamento de Cardiología de la Clínica Universitaria Benjamin Franklin en Berlín, Alemania.

La base de datos contiene 549 registros de 290 sujetos. Cada sujeto está representado por uno a cinco registros. No hay sujetos numerados. Cada registro incluye 15 señales medidas simultáneamente: las 12 derivaciones convencionales (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6) junto con los ECG de 3 derivaciones de Frank (vx, vy, vz). Cada señal se digitaliza a 1000 muestras por segundo, con una resolución de 16 bits en un rango de  $\pm 16,384$  mV. A pedido especial de los contribuyentes de la base de datos, los registros pueden estar disponibles a frecuencias de muestreo de hasta 10 KHz. [1]

### 3.3 Exploratory Data Analysis (EDA)

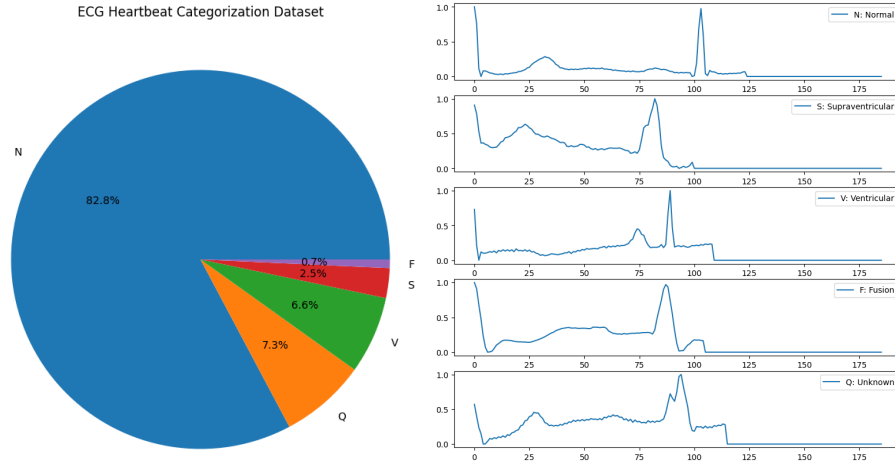


Fig. 1: Categorización del dataset

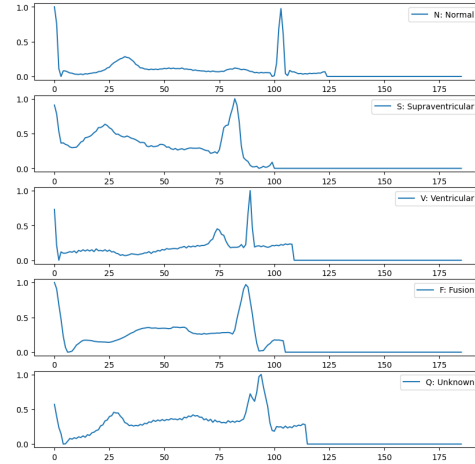


Fig. 2: Visualización de cada categoría

Este EDA tiene como objetivo comprender la estructura, distribución y patrones presentes en los datos de electrocardiogramas (ECG), que se utilizan para clasificar diferentes tipos de latidos cardíacos. Este análisis preliminar permite identificar características relevantes, evaluar el balance de clases y descubrir posibles correlaciones entre variables.

Esto nos permitira tomar decisiones informadas sobre la selección y preparación de características y desarrollar un modelo de clasificación mas efectivo.

En la Figura 1 podemos observar la distribución de los datos, y en la figura 2 podemos ver como se diferencian visualmente entre ellas las categorías, el objetivo del modelo es poder identificar a cual de estas categorías pertenece una lectura de ECG.

**Verificación de Valores Faltantes** Se confirmó que los datos están completos y no contienen valores nulos en ninguno de los cuatro conjuntos de datos (mitbih train, mitbih test, ptbdb abnormal, y ptbdb normal). Esto garantiza que no se requiere imputación ni eliminación de registros incompletos, lo que simplifica la preparación de los datos para el modelado posterior.

**Análisis de correlación** Se ha determinado que existe correlación entre los datos, esto es normal ya que los dataset provienen de señales de electrocardiogramas (ECG), que miden la actividad eléctrica del corazón en diferentes momentos del tiempo y representan mediciones continuas de la actividad cardíaca. Podemos observar el alto nivel de correlación de los datos en la Figura 3.

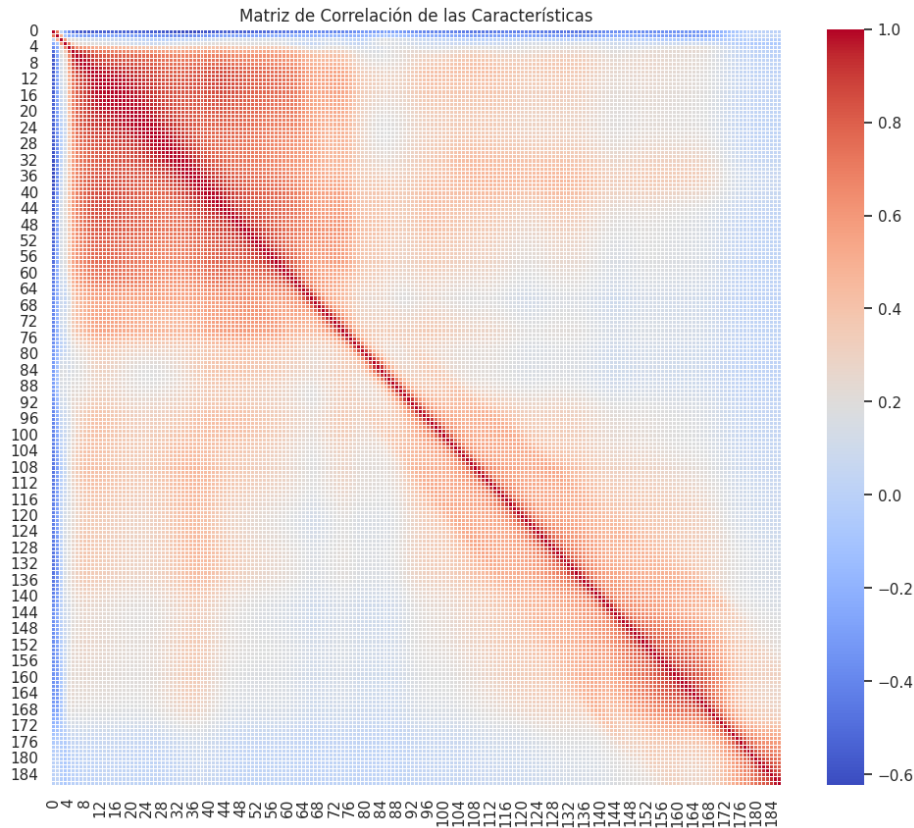


Fig. 3: Matriz de Correlación de las Características

Esta correlación es un aspecto esperado y puede ser de ayuda para capturar patrones complejos de la actividad cardíaca que pueden ser indicativos

de condiciones específicas, pero también puede llegar a causar problemas en el entrenamiento del modelo.

**Análisis de distribución de clases** Al observar el histograma de la figura 4 podemos identificar que el conjunto de datos está desbalanceado, es decir, algunas clases están representadas con mucha mayor frecuencia que otras.

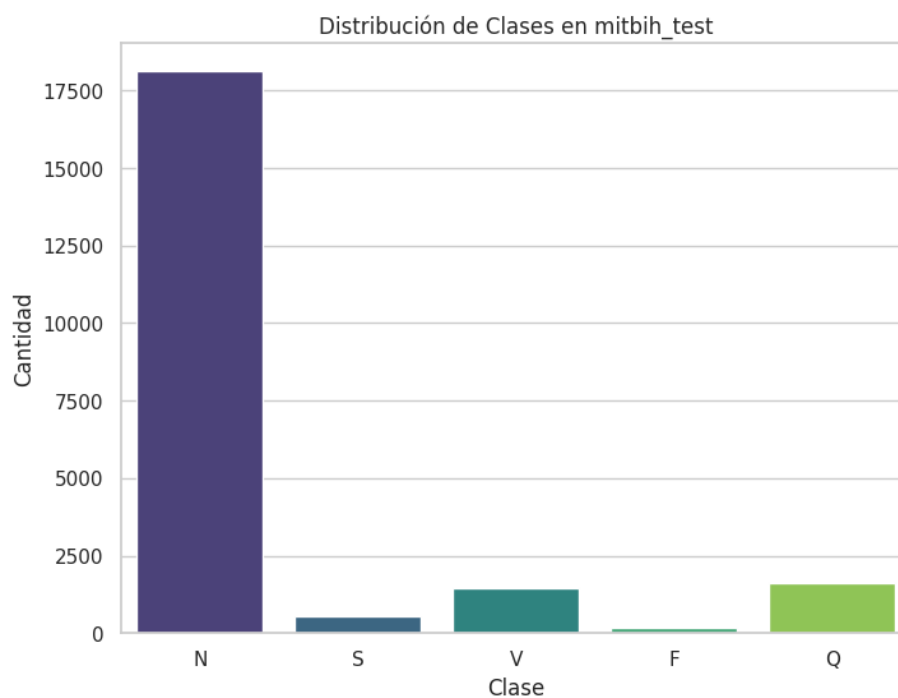


Fig. 4: Histograma del conjunto de datos

Este desequilibrio de clases puede afectar negativamente el rendimiento de nuestro modelo, evitando que pueda detectar con precisión todas las clases. Podrían ser necesarias técnicas de balanceo de clases, como la submuestreo de la clase mayoritaria o el sobremuestreo de la clase minoritaria.

## 4 Marco Teórico

### 4.1 Sobre el modelo

Para resolver este problema de categorización de eventos cardíacos, consideramos un enfoque de clasificación supervisado.

Dependiendo del modelo es si utilizaríamos librerías como TensorFlow, Keras, PyTorch, scikit-learn. Para evaluar los modelos utilizaremos métricas como precisión, recall, F1-score, matriz de confusión y así evaluar el rendimiento del modelo en la clasificación de eventos cardíacos. A priori, no podemos definir cual es el modelo más adecuado, requerimos de implementar los modelos y entrenarlos lo mejor posible. Será al final que concluiremos cuál es el más adecuado para resolver este problema con los datos brindados.

Dada la naturaleza de los datos, los cuales son numéricos, fue que consideramos dichos modelos para que sean compatibles.

Entre los modelos que consideramos desarrollar en este proyecto son:

- SVC
- Random Forest Classifier
- Logistic Regression
- Decision Tree Classifier
- AdaBoost Classifier

Para la solución del problema se plantea el entrenamiento de dos modelos. El primer modelo es entrenado para identificar únicamente dos posibles estados: el paciente está enfermo o está sano, de acuerdo a los resultados del ECG. En caso de que el primer modelo identifique que la persona presenta una enfermedad, entra en juego el segundo modelo, que se encarga de identificar específicamente qué enfermedad presenta la persona. Podemos observar una representación gráfica de la metodología en la figura 5.

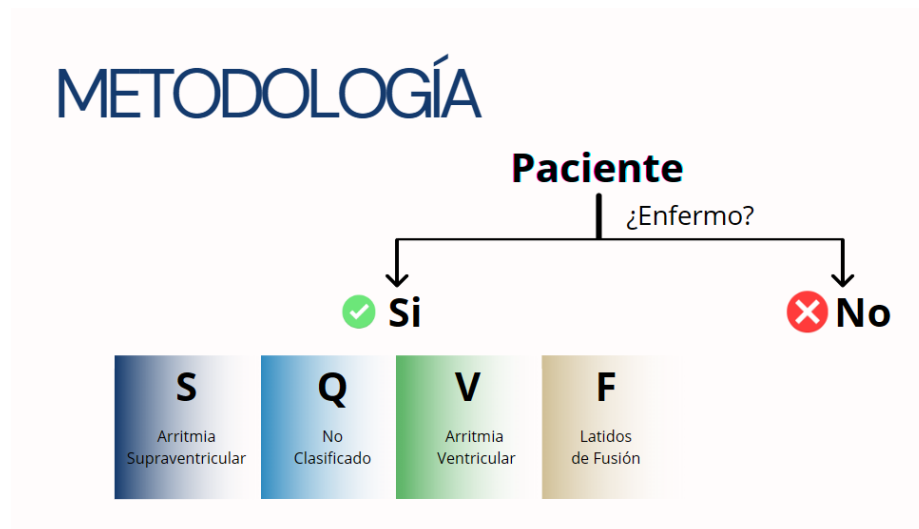


Fig. 5: Metodología utilizada

## 4.2 Procesamiento de los datos

El primer paso antes de entrenar los modelos es verificar la integridad de los datos. Se realizó la verificación de si hay datos faltantes o inconsistentes, y se llevó a cabo la búsqueda de valores atípicos. En caso de ser necesario, se realiza una codificación de datos, convirtiendo las variables categóricas en numéricas utilizando técnicas como la codificación one-hot o la codificación por etiquetas.

En el dataset encontramos que todos los valores son numéricos y tienen un rango de entre 0 y 1, al no haber variables categóricas, no es necesario realizar codificación. Los datos son consistentes y están completos.

## 4.3 Balanceo de muestras

Al observar el histograma de la figura 4 en la sección del EDA podemos observar que el conjunto de datos está desbalanceado, por lo que es necesario seguir un proceso para equilibrar los datos. Los modelos elegidos realizan este proceso durante el entrenamiento mediante la técnica de weight loss.

La técnica de weighted loss ajusta el peso de las clases en el cálculo de la función de pérdida durante el entrenamiento del modelo. Esta técnica asigna más peso a las clases minoritarias y menos a las clases mayoritarias, lo que hace que el modelo preste más atención a las clases con menos representaciones y trate de reducir el error para todas las clases de manera más equitativa. Una gran ventaja de esta técnica es que no requiere modificar los datos, a diferencia de las técnicas de upsampling o downsampling.

## 4.4 Selección de características

Para identificar las características más relevantes del modelo, se utilizó el método de selección de características RFE (Recursive Feature Elimination). Esta técnica funciona mediante un proceso iterativo que elimina, de forma recursiva, las características menos importantes hasta quedarse con un subconjunto de las más significativas. Establecimos que, del total de características que teníamos, el método nos retornara las 54 más importantes, lo que constituye aproximadamente el 30% del total de características.

## 4.5 Entrenamiento y validación de los modelos

**Modelo 1: Clasificador de persona enferma o no enferma** Una vez determinadas las características, entrenamos los modelos utilizando el conjunto de datos de entrenamiento.

Para el modelo que identifica si la persona está enferma o no, el mejor modelo fue AdaBoostClassifier. Los resultados pueden observarse en la figura 6.

Se eligió este modelo porque tiene una precisión más equilibrada entre ambas clases, ya que otros modelos, como Random Forest, tienen una precisión de 0.97 para la clase 2 pero solo 0.60 para la clase 1. El modelo de AdaBoost nos da una precisión de 0.77 para la clase 1 y 0.90 para la clase 2. Otro factor de selección

	precision	recall	f1-score	support
0.0	0.77	0.72	0.74	803
1.0	0.90	0.92	0.91	2108
accuracy			0.86	2911
macro avg	0.83	0.82	0.82	2911
weighted avg	0.86	0.86	0.86	2911

Fig. 6: resultados de entrenamiento y prueba de AdaBoostClassifier Modelo 1

fue que AdaBoost también posee, en general, los mejores resultados respecto al recall, ya que cuenta con 0.92 para la clase 1, esto es importante ya que la clase 1 representa a las personas enfermas, por lo que el modelo tiene menos preobabilidades de diagnosticar a una persona enferma como sana.

**Modelo 2: Clasificador de enfermedades** Nuevamente entrenamos los modelos utilizando el conjunto de datos de entrenamiento correspondiente.

Para el modelo que identifica la enfermedad del paciente, el mejor modelo fue AdaBoostClassifier, los resultados pueden observarse en la figura 7.

	precision	recall	f1-score	support
1.0	0.85	0.83	0.84	556
2.0	0.83	0.89	0.86	1448
3.0	0.64	0.57	0.60	162
4.0	0.95	0.91	0.93	1608
accuracy			0.88	3774
macro avg	0.82	0.80	0.81	3774
weighted avg	0.88	0.88	0.88	3774

Fig. 7: resultados de entrenamiento y prueba de AdaBoostClassifier Modelo 2

Tras obtener los resultados, observamos que la mayoría de los modelos tuvieron problemas con una de las clases; casi todos los modelos obtuvieron una

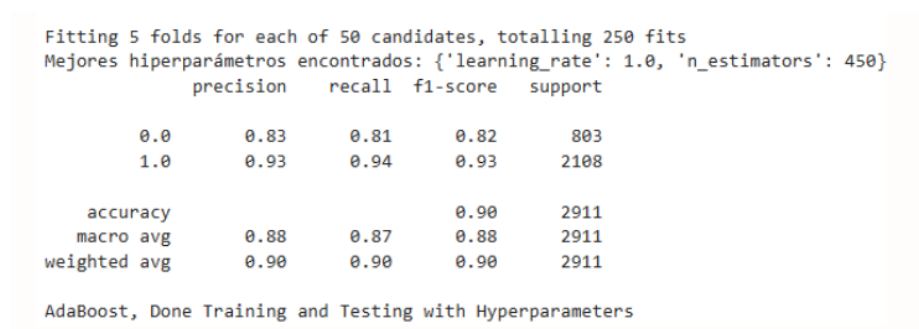


precisión de 0.50 o menos para la clase 3. A diferencia de ellos, AdaBoostClassifier obtuvo una precisión de 0.64, que, aunque no es ideal, es mejor que el resto de los modelos. En el resto de las clases, ningún modelo tuvo problemas en obtener precisiones mayores a 0.80.

#### 4.6 Ajuste de hiperparámetros

Ya que hemos elegido nuestros modelos, podemos ajustar los hiperparámetros para que el modelo se ajuste mejor a los datos e incrementar las capacidades de nuestro modelo. Para el ajuste de parámetros utilizamos la técnica de Grid Search, que básicamente busca la combinación óptima entre todas las posibilidades.

**Modelo 1: Clasificador de persona enferma o no enferma con hiperparámetros ajustados** En la figura 8 podemos observar los nuevos resultados tras entrenar y validar el modelo con hiperparámetros ajustados.



```
Fitting 5 folds for each of 50 candidates, totalling 250 fits
Mejores hiperparámetros encontrados: {'learning_rate': 1.0, 'n_estimators': 450}
```

	precision	recall	f1-score	support
0.0	0.83	0.81	0.82	803
1.0	0.93	0.94	0.93	2108
accuracy			0.90	2911
macro avg	0.88	0.87	0.88	2911
weighted avg	0.90	0.90	0.90	2911

AdaBoost, Done Training and Testing with Hyperparameters

Fig. 8: resultados de entrenamiento y prueba de AdaBoostClassifier ajustado - Modelo 1

Observamos un incremento en precisión para ambas clases: la clase 1 pasó de 0.77 a 0.83 y la clase 2 pasó de 0.90 a 0.93. También observamos un incremento en el recall: en la clase 1 pasó de 0.72 a 0.81 y en la clase 2 de 0.92 a 0.94.

Con estos resultados, podemos concluir que el ajuste de hiperparámetros sí aumentó el desempeño del modelo y permite obtener mejores resultados

**Modelo 2: Clasificador de enfermedades con hiperparámetros ajustados** En la figura 9 podemos observar los nuevos resultados tras entrenar y validar el modelo con hiperparámetros ajustados.

Observamos ligeros cambios en la precisión de cada clase, siendo el cambio más importante el incremento en la clase 3, de 0.64 a 0.69. En el caso del recall, también observamos ligeros cambios, siendo nuevamente la clase 3 la más relevante, con un incremento de 0.57 a 0.69.

```

Fitting 5 folds for each of 50 candidates, totalling 250 fits
Mejores hiperparámetros encontrados: {'learning_rate': 0.1, 'n_estimators': 400}

```

	precision	recall	f1-score	support
1.0	0.84	0.81	0.82	556
2.0	0.84	0.90	0.87	1448
3.0	0.69	0.69	0.69	162
4.0	0.96	0.92	0.94	1608
accuracy			0.88	3774
macro avg	0.83	0.83	0.83	3774
weighted avg	0.89	0.88	0.88	3774

```

AdaBoost, Done Training and Testing with Hyperparameters

```

Fig. 9: resultados de entrenamiento y prueba de AdaBoostClassifier ajustado - Modelo 2

Con estos resultados, podemos concluir que el ajuste de hiperparámetros, aunque en menor medida que en el modelo 1, sí aumentó el desempeño del modelo y permitió obtener mejores resultados.

#### 4.7 Pruebas y conclusiones

Las pruebas realizadas en ambos modelos demostraron que ambos clasificaron correctamente de acuerdo con los datos proporcionados. Aunque ambos modelos mostraron un buen rendimiento, es posible que exista un modelo aún mejor para la clasificación de enfermedades. Sin embargo, en este caso, seleccionamos AdaBoost debido a que mostró un desempeño más equilibrado entre todas las clases, ofreciendo una mejor consistencia en los resultados.

## References

1. Bousseljot, Ralf-D. PTB Diagnostic ECG Database. PhysioNet. (2004) <https://www.physionet.org/content/ptbdb/1.0.0/>
2. Soto-Estrada, Guadalupe, Moreno-Altamirano, Laura, and Pahua Díaz, Daniel. (2016). Panorama epidemiológico de México, principales causas de morbilidad y mortalidad. Revista de la Facultad de Medicina (México), 59(6), 8-22. Recuperado en 10 de septiembre de 2024, de [http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S0026-17422016000600008&lng=est&lng=es](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0026-17422016000600008&lng=est&lng=es).

## Contribuciones de los Autores

- **Grace Aviance:**
  - Explicación de ECG

- Introducción
- Descripción del Dataset
- **Carlos Sánchez:**
  - Entendimiento de los datos
  - Selección de Características
  - Entrenamiento y validación de los modelos
- **Fabian Lioner:**
  - Sobre el modelo
  - Procesamiento de los datos
  - Balanceo de muestras
- **Samuel Padilla:**
  - Abstract
  - Exploratory Data Analysis (EDA)
  - Ajuste de hiperparámetros
  - Pruebas y conclusiones