ECG Heartbeat Classification: A Deep Transferable Representation

Cynthia Cristal Quijas Flores[1] A01655996, Alejandro Sanchez Flores[2] A01662783, Carlos Adrián Palmieri Álvarez[3]A01635776, and Dabria Camila Carrillo Meneses[4]A01656716

Instituto Tecnológico y de Estudios Superiores de Monterrey - Campus GDL

Abstract. En el siguiente proyecto se analiza información de latidos cardiacos usando la base de datos "ECG Heartbeat Categorization Dataset", el cual se compone de más de cien mil muestras de señales de electrocardiogramas que se clasifican en cinco categorías principales. Para el análisis los datos se dividen en un conjunto de entrenamiento y otro de prueba con el fin de entrenar y después evaluar modelos de clasificación propuestos. El estudio de datasets de este tipo tienen grandes aplicaciones en el mundo de la salud y tecnología, desarrollando modelos predictivos para la detección de anomalías y problemas cardiacos como arritmias. Además de estas anomalías se puede usar para diferentes objetivos teniendo mejores implementaciones y por ende, mejor desempeño de modelos, todo esto teniendo modelos más sofisticado y resultados de calidad con sinfín de aplicaciones.

Keywords: Electro-cardiograma · Análisis · Predicción · Clasificación

1 Introducción

El proyecto tiene relevancia en México debido a que, alrededor de 220 mil personas fallecieron por enfermedades cardiovasculares en 2021, los cuales, pueden ser reducidos considerablemente con la correcta identificación temprana y controlar factores de riesgo como lo son el tabaquismo, la obesidad, el estrés y diabetes; estos mismos factores de riesgo se pueden mitigar con una correcta vida diaria, buena alimentación y chequeos médicos frecuentes. Una implementación que ayude a los cardiólogos a identificar con eficiencia y eficacia pudiera beneficiar a la reducción de fallecimientos en la república mexicana, incluso, en el futuro se podrían implementar dispositivos que pudieran tener las personas en sus hogares y que les facilitara la detección de anomalías cardiacas (Gobierno de México, 2024).

2 Metodología Implementada

La metodología implementada a lo largo del proyecto es llamada CRISP-DM o Cross-Industry Standard Process for Data Mining, de la compañia IBM; la metodología consiste en diferentes fases para poder lograr un proceso de minería de datos con estándares mundiales y que producen valor (IBM Corp., s.f.).

2.1 Comprensión del negocio

En esta primer fase de la metodología consiste en conocer los conceptos básicos del negocio, empresa u organización a la que se está enfocando el proyecto, también considerar los objetivos comerciales para poder obtener resultados de valor al negocio (IBM Corp., s.f.).

2.2 Comprensión de los datos

Una vez entendidos los objetivos, los interesados del proyecto y entendido el plan de trabajo a realizar, la fase que continúa es la comprensión de los datos. Aquí es necesario entender los conceptos de los datos, sus descripciones, explorar los datos, verificar su procedencia y revisar la calidad de los datos (IBM Corp., s.f.).

2.3 Preparación de los datos

En esta fase de la metodología ya se empiezan a emplear técnicas de ciencia de datos para poder introducirlos al algún modelo de aprendizaje de máquina y que arroje los mejores resultados posibles. Aquí se emplean técnicas de transformación, limpieza, construcción de nuevos datos y formateo de los mismos (IBM Corp., s.f.).

2.4 Modelado

La fase de modelado implica la construcción y generación de modelos e implementando técnicas de mejora de los mismos (IBM Corp., s.f.).

2.5 Evaluación

Esta fase evalúa el paso anterior, el cuál busca identificar aquellos modelos que tuvieron un mejor rendimiento (IBM Corp., s.f.).

2.6 Despliegue

Esta es la fase en se planifica cómo se va a implementar los modelos para que el usuario final pueda hacer uso de los mismos y cómo se llevará un control y mantenimiento de los mismos (IBM Corp., s.f.).

3 Descripción del dataset

El Dataset Heartbeat Categorization es un conjunto de datos creado para el análisis y clasificación de conjuntos de latidos cardíacos obtenidos a través de electrocardiogramas realizados a cierto número de personas. Este dataset es específicamente útil en el ámbito de la medicina y la investigación ya que permite la creación de modelos que pueden detectar arritmias u otras anomalías en los

latidos del corazón. Los datos reflejan las señales del corazón, registrando la intensidad de los latidos a lo largo de un periodo de tiempo. Esta información se usa para entrenar y evaluar modelos de machine learning con el propósito de clasificar estas anomalías cardiológicas.

Una de las principales características de este dataset es que contiene una amplia variedad de muestras de latidos cardíacos, los cuales están representados mediante una serie de valores que describen su amplitud a lo largo del tiempo. Estos latidos cardíacos están categorizados en cinco clases diferentes:

- Normal (N): Latidos cardíacos normales
- Supraventricular (S): Latidos cardíacos ectópico supraventriculares.
- Ventricular (V): Latidos cardíacos ectópicos ventriculares.
- Fusion (F): Latidos cardíacos de fusión.
- Unknown (Q): Latidos cardíacos que no pertenecen a ninguna de las categorías anteriores.

El dataset se divide principalmente en dos partes: el conjunto de entrenamiento y el conjunto de prueba. El primero se utiliza para entrenar los modelos, permitiéndoles aprender patrones y características de los diferentes tipos de latidos. Por otro lado, el conjunto de prueba se emplea para evaluar el rendimiento del modelo, midiendo su capacidad de generalización y precisión en la clasificación de nuevos datos. Este tipo de evaluación es esencial para validar el comportamiento del modelo y asegurar su eficacia en la detección de anomalías cardiacas.

Existen numerosos conjuntos de datos diseñados específicamente para su uso en investigaciones de machine learning, y este conjunto no es la excepción. Su principal objetivo es categorizar y distinguir diferentes tipos de latidos cardíacos mediante el entrenamiento y evaluación de modelos de clasificación. Las aplicaciones de estos modelos son vastas, ya que pueden utilizarse para desarrollar herramientas médicas capaces de detectar de manera rápida y precisa una amplia gama de anomalías cardíacas.

4 Exploración de datos

Se realizó un análisis exploratorio de datos sobre el "ECG Heartbeat Categorization Dataset", el cual contiene señales de electrocardiogramas (ECG) categorizadas en diferentes tipos de latidos cardíacos. El análisis fue llevado a cabo para comprender la estructura de los datos y las relaciones entre las características, con el fin de preparar el dataset para un modelado predictivo.

4.1 Distribución test-train

El conjunto de datos proporcionados por el MIT cuenta con una previa separación del conjunto de datos, conteniendo 87,553 registros con 187 características y una columna de "target" la cuál contiene una de las 5 clases que contiene el

conjunto de datos. El conjunto de datos test cuenta con 21,891 registros y sin etiquetas con el fin de poder evaluar el modelo correctamente. Es decir que está dividido en 80 porciento de datos en el conjunto de datos de train y alrededor del 20 porciento en el conjunto de datos test.

4.2 Distribución de clases

Se realizó un análisis de la distribución de las diferentes categorías de latidos cardíacos en el conjunto de datos de entrenamiento mediante gráficos de barras. Este análisis es esencial para comprender la distribución y balance de cada clase con el fin de implementar estrategias adecuadas en caso de desequilibrio entre las clases.

Primero, se contabilizó el número de instancias para cada clase en los conjuntos de datos de entrenamiento y de prueba. Los resultados obtenidos se presentan a continuación en la Figura 1.

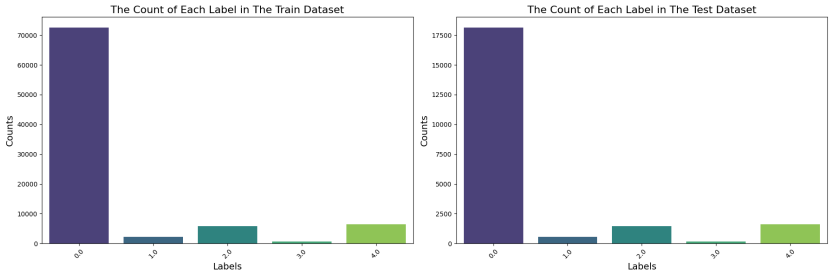

Fig. 1: Distribución de clases en el conjunto de datos de entrenamiento.

En donde se puede observar que la clase que tiene mayor presencia para ambos datasets es el "Normal". Esto podría generar que haya un sobre ajuste en la predicción de clases al momento de entrenar un modelo predictivo.

5 Balanceo de datos

Se realizó el balanceo del conjunto de datos, el cual originalmente presentaba una distribución desbalanceada entre las clases, teniendo una presencia significativamente mayor la clase normal.

Se verifica si existen valores faltantes en el conjunto de datos y se presenta la distribución original de las clases, lo que permite observar si alguna clase está sobrerrepresentada en comparación con las demás. Para corregir este desequilibrio, se utilizó la técnica de Random UnderSampling del paquete imblearn, que reduce la cantidad de muestras de las clases mayoritarias al mismo nivel de las clases minoritarias. Este proceso asegura que todas las clases tengan la misma

cantidad de observaciones, permitiendo un entrenamiento más equilibrado de los modelos de clasificación.

Finalmente, los resultados obtenidos tras haber aplicado la técnica seleccionada para el balanceo de datos fue que se tuvieron 641 registros para cada clase, así asegurándonos de que los modelos que se fueran a entrenar no tuvieran sesgo hacia ninguna clase en particular. A continuación se muestra el antes y el después de las clases con el balanceo.

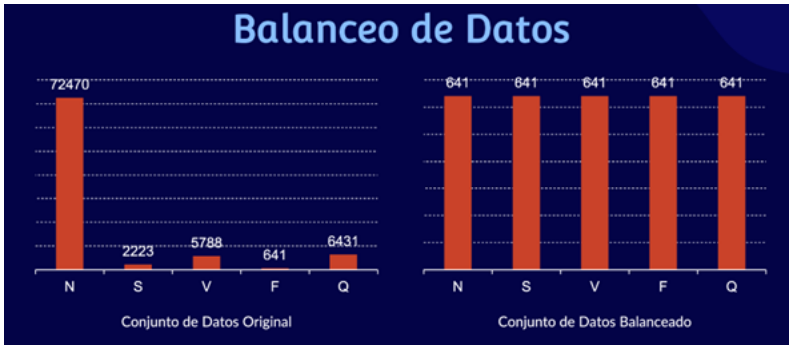

Fig. 2: Clases con y sin balanceo de datos

6 Modelo

6.1 Tipo de modelo que se requiere para resolver el problema del reto

En el análisis de señales de electrocardiogramas (ECG) con el objetivo de clasificar el tipo de enfermedad, los modelos de clasificación se presentan como herramientas altamente adecuadas debido a la naturaleza del problema. Las señales de ECG son datos temporales y secuenciales que capturan la actividad eléctrica del corazón, y cada patrón en estas señales puede estar asociado con distintos tipos de anomalías o enfermedades cardíacas. La columna que clasifica el tipo de enfermedad proporciona etiquetas categóricas que corresponden a diferentes condiciones patológicas, haciendo que el problema sea inherentemente una tarea de clasificación. Los modelos de clasificación, como los clasificadores basados en árboles de decisión, máquinas de soporte vectorial, redes neuronales o modelos de ensamble, son diseñados para aprender a distinguir entre categorías específicas basándose en características extraídas de las señales de ECG. Además, estos modelos pueden manejar la complejidad y variabilidad inherentes a los datos de ECG, proporcionando predicciones precisas y útiles para el diagnóstico médico. Al ajustar estos modelos con datos etiquetados, se pueden identificar patrones que ayudan a clasificar correctamente nuevas señales de ECG, facilitando un

diagnóstico efectivo y oportuno. En consecuencia, el uso de modelos de clasificación no solo es apropiado sino esencial para la interpretación precisa de las señales de ECG y la identificación de diferentes tipos de enfermedades cardíacas.

6.2 Tipo de datos tienen y cuáles modelos son compatibles con ellos

Datos numéricos categóricos se pueden emplear modelos cuadraticos y no lineales, probabbilisticos, geometricos y de ensamble clasificadoress poeque buscamos predecir entre clases El dataset contiene datos numéricos y categóricos, debido a esto existen diferentes modelos que podrían ser usados para resolver el problema planteado. Algunos de estos modelos son: cuadráticos,

6.3 Modelos que puedan usar para resolver el problema

Para poder resolver el problema del reto se pueden usar múltiples implementaciones de modelo, pero en este caso se propusieron los siguientes:

- Random forest
- SVM
- Naive Bayes
- Gradient Boosting
- MLP Classifier
- KNN
- Logistic Regression
- Decision Tree
  Estos modelos de clasificación son útiles para lograr el objetivo, de esta manera se eligieron los cuatro mejores en general para aplicarlos en el proyecto. A continuación se muestra la tabla con los resultados obtenidos tras evaluar a los modelos con el dataset balanceado.

MODELO	ACCURACY	PRECISION	RECALL	F1-SCORE
Logistic Regression	74.85%	74.78%	74.85%	74.72%
KNN	83.65%	83.77%	83.65%	83.66%
Decision Tree	79.31%	79.28%	79.31%	79.23%
Random Forest	88.64%	89.14%	88.64%	88.76%
SVM	77.60%	78.58%	77.60%	77.84%
Naive Bayes	28.55%	39.54%	28.55%	21.28%
Gradient Boosting	85.34%	85.71%	85.33%	85.45%
MLP Classifier	86.15%	86.18%	86.15%	86.11%

Table 1: Resultados de la evaluación de modelos con datos balanceados

6.4 Entrenamiento y prueba de modelos

Para probar la precisión y exactitud de los modelos, se utilizaron los datos proporcionados en el archivo "mitbihtrain.csv", a estos mismos, se realiza un submuestreo aleatorio para evitar generar un sobreajuste sobre los datos, también, se dividen en conjunto de entrenamiento y prueba, 80% y 20% respectivamente.

Una vez preparados los modelos, se obtuvieron que los tres mejores modelos fueron los siguientes:

– RandomForestClassifier()

```
-------- Evaluando RF --------
F1 Score: 0.8741
              precision    recall  f1-score   support

         0.0       0.81      0.83      0.82       158
         1.0       0.84      0.89      0.86       118
         2.0       0.93      0.90      0.91       124
         3.0       0.90      0.88      0.89       121
         4.0       0.98      0.96      0.97       120

    accuracy                           0.89       641
   macro avg       0.89      0.89      0.89       641
weighted avg       0.89      0.89      0.89       641
```

Fig. 3: Reporte de Clasificación para modelo de Bosques Aleatorios

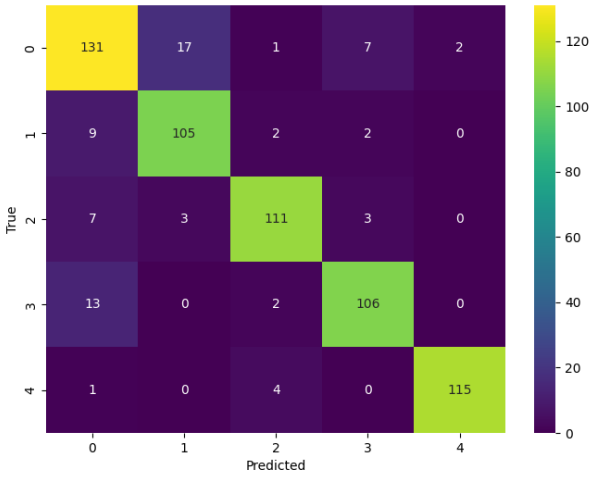

Fig. 4: Matriz de confusión para modelo de Bosques Aleatorios

– GradientBoostingClassifier()

```
-------- Evaluando GB --------
F1 Score: 0.8490
              precision    recall  f1-score   support

         0.0       0.75      0.75      0.75       158
         1.0       0.84      0.83      0.83       118
         2.0       0.83      0.85      0.84       124
         3.0       0.88      0.87      0.87       121
         4.0       0.93      0.93      0.93       120

    accuracy                           0.84       641
   macro avg       0.84      0.85      0.84       641
weighted avg       0.84      0.84      0.84       641
```

Fig. 5: Reporte de Clasificación para modelo de GradientBoosting

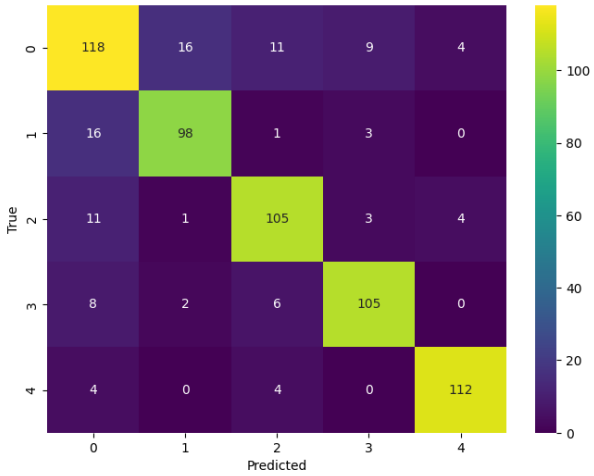

Fig. 6: Matriz de confusión para modelo de GradientBoosting

– MLPClassifier(hidden_layer_sizes=(100, 100), max_iter=1000)

```
-------- Evaluando MLP --------
F1 Score: 0.8564
              precision    recall  f1-score   support

         0.0       0.81      0.80      0.80       158
         1.0       0.80      0.86      0.82       118
         2.0       0.83      0.86      0.85       124
         3.0       0.94      0.89      0.92       121
         4.0       0.98      0.93      0.96       120

    accuracy                           0.86       641
   macro avg       0.87      0.87      0.87       641
weighted avg       0.87      0.86      0.87       641
```

Fig. 7: Reporte de Clasificación para modelo de Multi-layer Perceptron

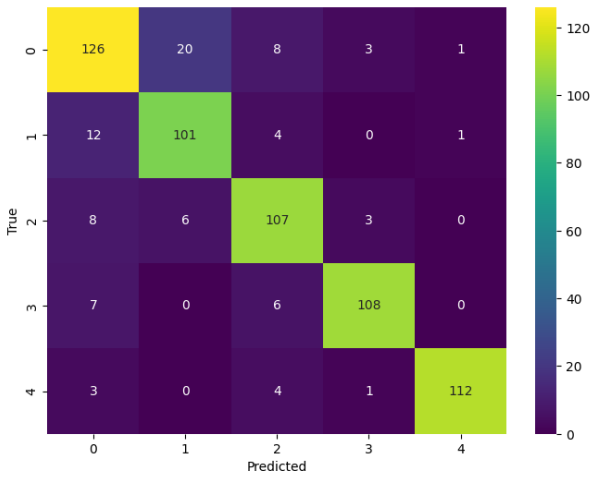

Fig. 8: Matriz de confusión para modelo de Multi-layer Perceptron

6.5 Decisiones tomadas sobre los modelos

A partir de los cuatro mejores modelos, se decidió que se van a volver a modelar aplicando técnicas de selección de hiperparámetros para cada uno.

Los 8 modelos probados para realizar la clasificación, también se probó mejorar una estructura de red neuronal. Esta llegó a tener resultados bastante elevados pero preocupantes, a pesar de haber hecho la la división de los datos de entrenamiento, validación y prueba; se estaba realizando un sobre ajuste a los datos debido a la complejidad del modelo. A continuación se presentan los resultados de la red neuronal con diferentes técnicas:

Tamaño de capas ocultas	**Tamaño de neuronas por capa**
3	128, 64, 32
4	128, 64, 32, 16
3	128, 50, 20
5	128, 100, 70, 50, 30
6	128, 100, 80, 60, 40, 32
7	128, 100, 80, 60, 40, 32, 16

Table 2: Búsqueda de arquitecura de la red neuronal

Tamaño batch	**Épocas**	**Optimizador**
32	30	Adam
64	50	
128	70	SDG
256	100	

Table 3: Búsqueda de hiperparámetros para red neuronal, usando validación cruzada con 3 folds

Al final de la búsqueda de hiperparámetros y una búsqueda de la mejor configuración para la estructura de la red neuronal, se determinó que la red neuronal está dada por la siguiente estructura y parámetros: Función de activación para capas escondidas: ReLU; optimizador: Adam; Épocas: 3000; Batch size: 128. 200, 128, 100, dropout(.5), 60, 40, dropout(.5), 32, 16, 5.

Este modelado generó los siguientes resultados:

Clase	PRECISION	RECALL	F1-SCORE
0	99%	99%	99%
1	88%	74%	81%
2	95%	93%	94%
3	72%	75%	73%
4	99%	97%	98%

Table 4: Resultados de la red neuronal ajustada sin técinas de balanceo de datos.

Gradient Boosting Para este modelo se ajustaron los hiperparámetros usando el dataset balanceado usando el random undersampler, los hiperparámetros que se ajustaron fueron learning rate y n estimators.

El proceso de optimización emplea la técnica de validación cruzada estratificada con 5 particiones, lo que garantiza que cada subconjunto tenga una proporción similar de clases. En cada iteración, se entrena el modelo con un subconjunto de datos y se evalúa su desempeño en otro subconjunto, calculando la precisión promedio. Este ciclo se repite para cada combinación de hiperparámetros, y se guarda aquella configuración que maximiza la precisión.

Los resultados obtenidos hasta el momento durante la búsqueda de hiperparámetros se presentan en la Tabla 6. En cada iteración, se han evaluado diferentes combinaciones del número de estimadores y la tasa de aprendizaje utilizando validación cruzada estratificada. Hasta ahora, se han completado varias combinaciones, mostrando mejoras significativas en la precisión a medida que se ajustan los hiperparámetros. Sin embargo, el proceso se mantuvo en curso durante 7337 minutos. A continuación, se detallan los resultados parciales obtenidos:

n_estimators	learning_rate	Precisión (ACC)
50	0.010	0.8807
50	0.031	0.9340
50	0.052	0.9477
50	0.073	0.9543
50	0.094	0.9574
50	0.073	No terminó

Table 5: Resultados parciales de la búsqueda de hiperparámetros utilizando Gradient Boosting

Cabe destacar que de los cuatro modelos que se eligieron este es probablemente el que necesita de recursos computacionales mayores que los demás, por lo que era esperado que el tiempo de ejecución fuera el mayor. Por este último motivo, se permitió la experimentación de una búsqueda de hiperparámetros robusta. Hasta el momento en que se detuvo la ejecución estaba alcanzando unos resultados favorables, mejorando de un 85% en métricas como accuracy, recall,

f1-score, precision, hasta 95%, el cual podría considerarse todavía un modelo aceptable sin caer en el overfitting. Finalmente, nada más mencionar que este modelo quedó descartado para ser el final por el tiempo que se necesitó para ejecutarlo, pero esto no quiere decir que sus resultados sean desfavorables.

Random Forest Para este experimento se optó por este modelo de clasificación, el cual nos proporciona de diversas ventajas en conjuntos de datos robustos como este. Una de ellas, es que al encontrarnos con un conjunto de datos muy desbalanceado, Random Forest incorpora una técnica específica para lidiar con el desequillibrio entre clases y de esta forma evitar el sesgo hacia la clase mayoritaría.

Para cada uno de los árboles de nuestro bosque, Balanced Random Forest reequilibra las clases tomando subconjuntos de muestra aleatorios con reemplazo, asegurando que haya un número igual de instancias de cada clase. Este proceso reequilibra las clases y mejora la capacidad del modelo para aprender de todas las clases, incluidas las minoritarias.

Al reentrenar múltiples árboles con esta técnica, Balanced Random Forest logra un mejor rendimiento al evitar que las clases mayoritarias dominen las predicciones, mejorando la precisión y el recall para las clases menos representadas. Esto resulta en un modelo más robusto y equitativo para escenarios de desbalance como este, donde la correcta clasificación de las clases minoritarias es crítica para el éxito del sistema.

n_estimators	**max_depth**	**min_samples_split**	**min_samples_leaf**	**bootstrap**
100	10	2	1	True
200	20	10	5	
500	None	20	10	False

Table 6: Búsqueda de hiperparámetros óptimos para BalancedRandomForest

KNN Para este modelo, se mantuvo la misma dinámica de implementación de los demás. Se entrenó un modelo de clasificación basado en el algoritmo K-Nearest Neighbors (KNN). Se separan las variables predictoras (X) de la variable objetivo (y) para posteriormente aplicar validación cruzada para evaluar el rendimiento del modelo KNN utilizando diferentes valores del hiperparámetro k. Para cada valor de k, se entrena el modelo, se generan predicciones y se calcula la precisión promedio, almacenando los mejores resultados. Finalmente, el código visualiza la relación entre k y la precisión, entrena un modelo con el valor óptimo de k y evalúa su rendimiento. Algunos resultados de los mejores k son:

- —- k = 1 ACC: 0.9761515661152014
- —- k = 2 ACC: 0.9746553343756087
- —- k = 3 ACC: 0.9752492422075824

– —- k = 3 ACC: 0.9752492422075824
– —- k = 5 ACC: 0.9736387991544257

7 Resultados

En el proyecto se obtuvieron resultados que muestran el rendimiento del modelo de clasificación aplicado al conjunto de datos de latidos cardíacos. Se generó una Matriz de Confusión que permite visualizar el comportamiento del modelo para cada una de las clases de latidos donde obtenemos información importante. La clase 0 (latidos normales) fue clasificada correctamente en la mayoría de los casos, con 65,001 predicciones correctas; sin embargo, el modelo tuvo errores al confundir algunos latidos normales con las clases 1, 2 y 3. Específicamente, 2,543 latidos normales fueron clasificados erróneamente como clase 1, 1,284 como clase 2 y 3,194 como clase 3. Para las clases 2 y 4, que corresponden a latidos ectópicos y de fusión respectivamente, el modelo también mostró un buen desempeño, aunque con errores menores en la predicción de algunas clases.

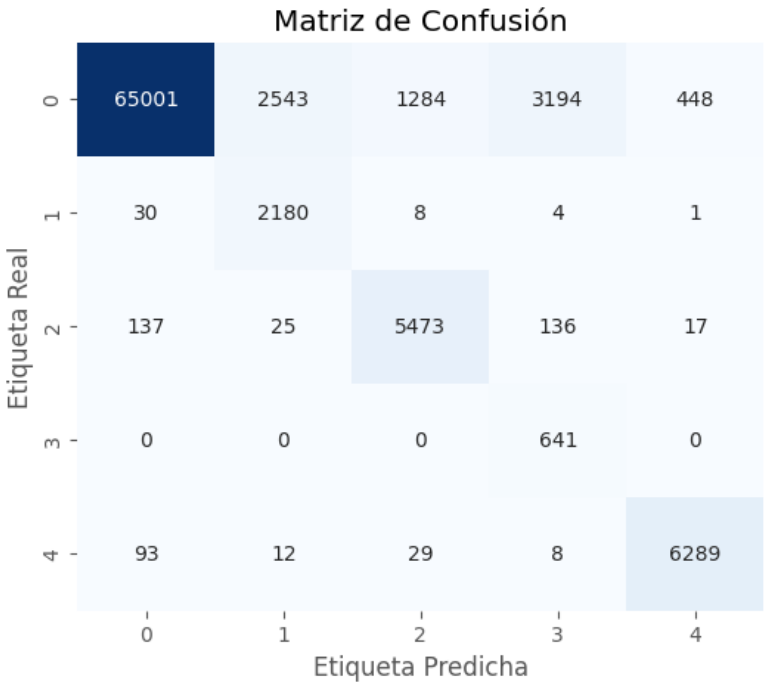

Fig. 9: Matriz de confusión del modelo

Se propuso la Curva Roc como otra manera de evaluación al modelo, esta proporciona una medida general del rendimiento de clasificación. En este caso,

la curva que se obtuvo muestra una capacidad de discriminación muy buena, con un AUC (Área bajo la curva) de 1.00, esto indica que el modelo es capaz de distinguir entre las diferentes clases de latidos con una precisión muy alta, lo que comprueba su gran utilidad en la detección de anomalías cardíacas.

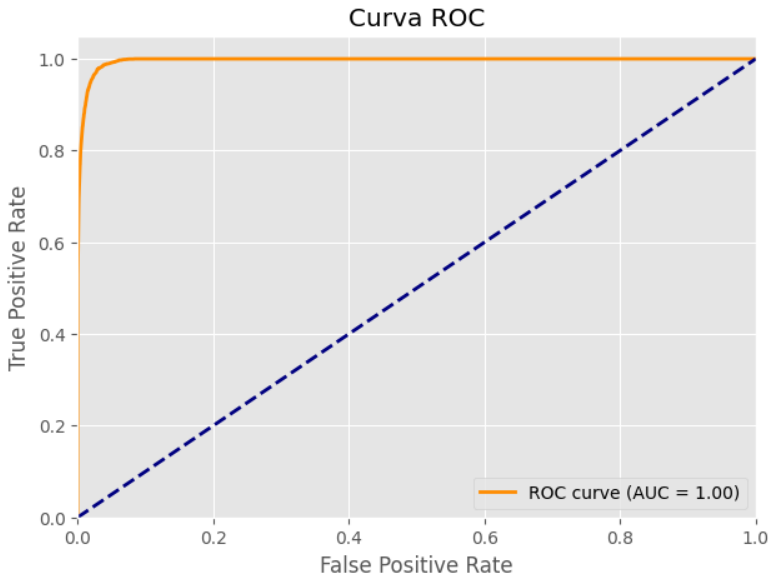

Fig. 10: Gráfico Curva Roc del modelo

Con base en los resultados de evaluación anteriores, podemos observar que el modelo tiene algunas confusiones mostradas en la matriz de confusión, a pesar de eso, el modelo logra un muy buen rendimiento en términos de su capacidad para clasificar correctamente los latidos en las cinco categorías correspondientes.

8 Conclusión

Se puede concluir que en este proyecto se logró proporcionar un análisis detallado de la clasificación de latidos cardíacos utilizando el conjunto de datos "ECG Heartbeat Categorization Dataset". Separando los datos en conjuntos de entrenamiento y prueba, se logró entrenar y evaluar diversos modelos de clasificación, obteniendo buenos resultados. El estudio e implementación de estos modelos tiene un impacto significativo en el campo de la salud y la tecnología, permitiendo el desarrollo de herramientas para la detección de anomalías cardíacas. Aunado a lo anterior, estos modelos pueden ser adaptados para múltiples propósito, a medida que se desarrollen modelos más sofisticados y precisos, las posibles aplicaciones en el ámbito médico seguirán creciendo, contribuyendo a un mejor diagnóstico y tratamiento de todo tipo de enfermedades.

9 Aportaciones

Cynthia Quijas:

Algunas de las aportaciones que hice para llegar a los resultados obtenidos fueron los siguientes: en primer lugar, la búsqueda exhaustiva de los mejores hiperparámetros para el modelo de Gradient Boosting. A través de un proceso de validación cruzada estratificada, evalué diversas combinaciones del número de estimadores y la tasa de aprendizaje. Esta optimización ha mostrado mejoras constantes en la precisión, alcanzando resultados prometedores. Sin embargo, la complejidad de este proceso también ha representado un desafío significativo, ya que el tiempo computacional requerido ha sido excesivo.

Otra contribución importante fue el uso de técnicas de balanceo de datos para corregir el problema de la distribución desigual de clases en el dataset. Utilicé el método de Random UnderSampling para reducir la representación de las clases mayoritarias y equilibrar el conjunto de datos, lo cual es fundamental en tareas de clasificación, ya que evita que el modelo esté sesgado hacia las clases más frecuentes como la que corresponde con la "normal".

Además, evalué un total de ocho modelos distintos, incluyendo algoritmos como Logistic Regression, Random Forest, SVM, y MLP Classifier. Cada uno de estos modelos fue evaluado en términos de precisión, F1 score, recall y precisión, utilizando validación cruzada con cinco particiones. Este análisis nos permitió comparar y entender cuáles algoritmos eran más adecuados para este problema de clasificación. A partir de estos experimentos, fue posible identificar qué enfoques ofrecían el mejor rendimiento sin necesidad de ajustar hiperparámetros inicialmente, lo que sirvió como base para futuras optimizaciones.

Carlos Palmieri:

Las aportaciones al proyecto de mi parte fueron la evaluación de la estructura de la metodología CRISP-DM e implementación de 8 modelos distintos para el problema de clasificación de nuestro conjunto de datos, entre ellos, modelos no lineales probabilísticos, geométricos y de ensamble. Construcción de una red neuronal probando más de 20 configuraciones de arquitectura. Aplicando validación cruzada para poder hacer una búsqueda de hiperparámetros y encontrando la mejor arquitectura para la red neuronal, arrogando métricas elevadas debido a la complejidad del modelo y esto puede involucrar un sobre ajuste del modelo en los datos.

Alejandro Sánchez:

Mis principales contribuciones al proyecto incluyeron la implementación del modelo de clasificación Random Forest y la optimización de sus hiperparámetros a través de una búsqueda exhaustiva. Utilicé un enfoque de GridSearchCV para ajustar parámetros clave como la cantidad de árboles, la profundidad máxima y el número mínimo de muestras en cada nodo, logrando identificar los valores óptimos. Gracias a esta búsqueda, el modelo Random Forest obtuvo los mejores resultados en cuanto a desempeño en las predicciones y desgaste computacional.

Además, generé diversas visualizaciones para evaluar el desempeño del modelo, incluyendo la matriz de confusión, la curva ROC y la curva de aprendizaje,

lo que nos permitió comprender mejor el comportamiento del modelo y su capacidad de generalización. Estas contribuciones ayudaron a optimizar el proceso de clasificación y lograron mejoras significativas en el rendimiento del modelo, superando a los otros enfoques considerados.

Dabria Carrillo:

Mis aportaciones se enfocaron en la implementación y optimización del algoritmo K-Nearest Neighbors (KNN) para la clasificación de los latidos cardíacos, selección de los hiperparámetros óptimos a través de un proceso de validación cruzada, evaluando diferentes valores de k para encontrar el óptimo, el cual diera como resultado una precisión más alta del modelo. También realicé un análisis del comportamiento del modelo en cada etapa, comprendiendo el funcionamiento de los algoritmos de clasificación y cómo los hiperparámetros afectan su rendimiento. Utilicé técnicas como la imputación de valores faltantes para mejorar los datos y propiciar que el modelo pudiera generalizar correctamente en el conjunto de prueba. Finalmente se pudo optimizar el modelo contribuir a la elección de los mejores modelos para la resolución de este reto.

References

1. Gobierno de México. (2024). Cada año, 220 mil personas fallecen debido a enfermedades del corazón. Recuperado de https://www.gob.mx/salud/prensa/490-cada-ano-220-mil-personas-fallecen-debido-a-enfermedades-del-corazon
2. IBM Corp. (s.f.). Introducción a CRISP-DM. IBM. Recuperado de https://www.ibm.com/docs/es/spss-modeler/saas?topic=guide-introduction-crisp-dm
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs, last accessed 2023/10/25