Instituto Tecnológico y de Estudios Superiores de Monterrey.

Campus Estado de México

Momento de Retroalimentación: Módulo 2 Uso de framework o biblioteca de aprendizaje máquina para la implementación de una solución.

ID3 Con Framework

Profesor

Jorge Adolfo Ramírez Uresti

Integrantes

Sebastián Espinoza Farías……. ITC | A01750311

Fecha de entrega: 10-09-2015

Introducción:

Durante este primer bloque de la concentración de Inteligencia artificial avanzada para la ciencia de datos, se desarrollaron dos implementaciones del algoritmo ID3. La primera siendo de manera manual, y la segunda se desarrolló utilizando los frameworks ya establecidos para que se pudiera comprender de mejor manera su funcionamiento e implementación, además de maneras de validar, ajustar y mejorar un algoritmo.

Al momento de realizar la primera implementación, se obtuvo un buen resultado, ya que este tenía un buen desempeño, sin embargo, pude notar con la segunda implementación, que al ser desarrollada con mucho más trabajo detrás, que esa implementación aún tiene un gran área de mejora. La segunda implementación fue mucho más sólida ya que no se pudo detectar con tanta facilidad sus puntos negativos.

Se espera que con esta tercera entrega, se pueda mejorar la segunda implementación haciendo uso de frameworks, ya que es un desarrollo que me parece mucho más aproximado a como se desarrollan soluciones en el mundo profesional.

Para esta tercer entrega ya se cuenta con algunos de los requerimientos que se piden, es por eso que se evidenciará durante el análisis donde y como se cumplen. Se utilizará el dataset the heart.csv, ya que los resultados pasados fueron buenos y sería bueno tratar de mejorarlos.

Desarrollo:

Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation).

En la presente implementación no se generó un tercer conjunto explícito de validación, ya que esta función se encuentra cubierta mediante el uso de validación cruzada dentro del procedimiento de GridSearchCV con el que ya contaba durante la segunda entrega.

El procedimiento fue el siguiente:

El conjunto de datos original se dividió en entrenamiento y prueba, los cuales generan un archivo csv mostrando exactamente que datos se utilizan para train y test. Así como una impresión en consola con sus características .

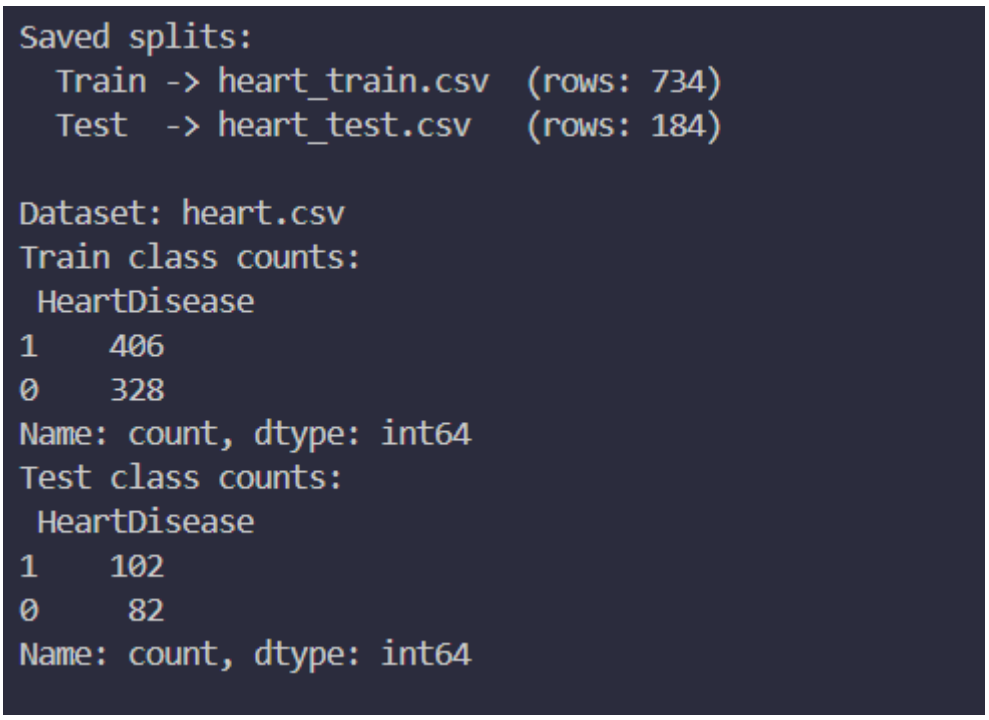


Imagen 1: Impresión en consola con características de train y test

El conjunto de entrenamiento fue utilizado por GridSearchCV, que a su vez aplica una estrategia de validación cruzada estratificada (StratifiedKFold). En cada iteración, los datos de entrenamiento se subdividen en conjuntos pequeños: algunos se utilizan para entrenar y uno para validar. De esta manera, cada instancia participa tanto en entrenamiento como en validación.

Se calculan métricas promedio en los pliegues de validación para comparar distintos hiper parámetros y seleccionar los que optimizan el desempeño del modelo.

Finalmente, el modelo se re-entrena con todos los datos de entrenamiento utilizando los mejores parámetros encontrados, y se evalúa en el conjunto de prueba, que permanece completamente independiente durante todo el proceso.

Este mecanismo asegura que la etapa de validación está presente de forma implícita, incluso de manera más robusta que con un único conjunto fijo de validación, ya que la validación cruzada permite aprovechar mejor los datos y reduce la varianza en la estimación del rendimiento del modelo.

Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto

En esta entrega se incorporó una nueva función dentro del código con el objetivo de diagnosticar de manera automática el grado de bias del modelo. Esta función nueva función recibe como entrada el accuracy obtenido en el conjunto de entrenamiento y, con base en umbrales definidos, clasifica el nivel de sesgo en alto, medio o bajo:

Bias alto: cuando el accuracy de entrenamiento es menor al 60 %. Esto significa que el modelo no logra aprender los patrones de los datos y presenta underfitting.

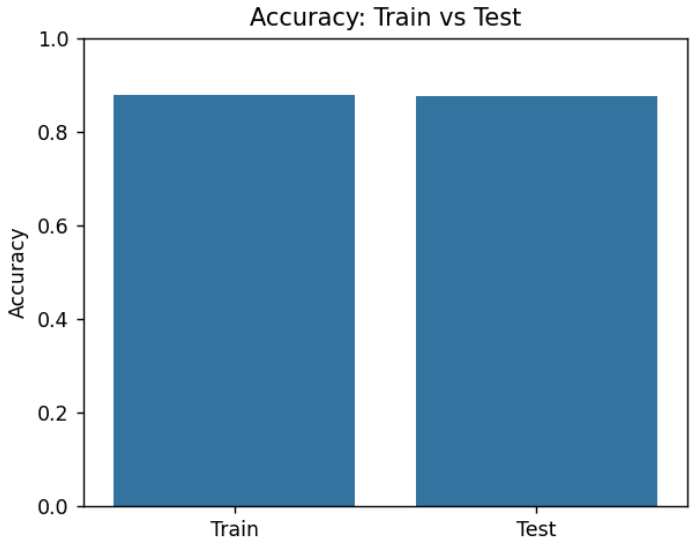
Bias medio: cuando el accuracy se encuentra entre 60 % y 80 %. En este caso, el modelo logra capturar parcialmente la información, pero aún presenta errores que sean obvios.

Bias bajo: cuando el accuracy es mayor o igual al 80 %. Esto indica que el modelo está aprendiendo adecuadamente la relación entre las variables y que el bias es bajo.

La función no solo asigna un nivel de bias, sino que también genera una explicación textual que interpreta los resultados y facilita el análisis. En este caso, usando el dataset Heart.csv, el modelo alcanzó un accuracy de entrenamiento de 0.877 y de prueba de 0.875, lo que llevó a diagnosticar un sesgo bajo, con la explicación:

“High training accuracy (≥80%). The model captures the relationship between variables well (low bias).”

Adicionalmente, se agregó la función plot_train_test_accuracy, que genera un gráfico de barras comparando el desempeño en entrenamiento y prueba. Esta visualización refuerza el diagnóstico, ya que permite observar si existe una gran diferencia entre ambos conjuntos (lo que sugeriría otros problemas como varianza u overfitting) o si los resultados son consistentes.

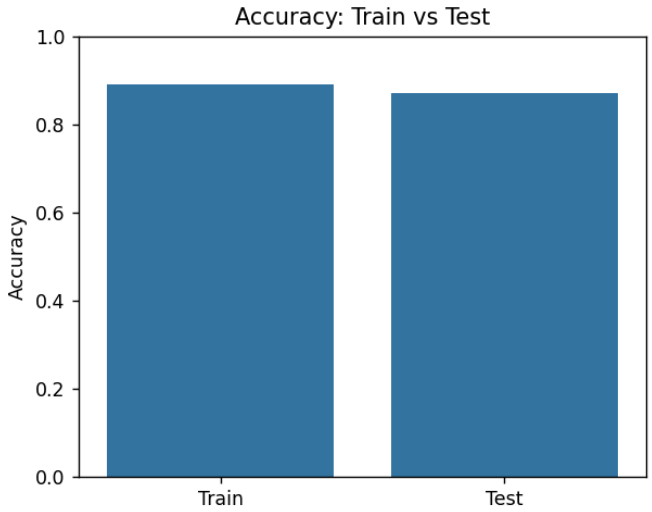



```
Accuracy
  Train: 0.8773841961852861
  Test : 0.875

Bias diagnosis: LOW
Explanation: High training accuracy (≥80%). The model captures the relationship between variables well (low bias).
```

Imagen 2: Gráfico de accuracy en train y test, impresión de resultado de bias 1

Se hizo una segunda evaluación del modelo, pero haciendo el split the train y test (70-30), lo que mejoró un poco el entrenamiento, per empeoró el desempeño en test. Sin embargo, no afectó el nivel de bias.

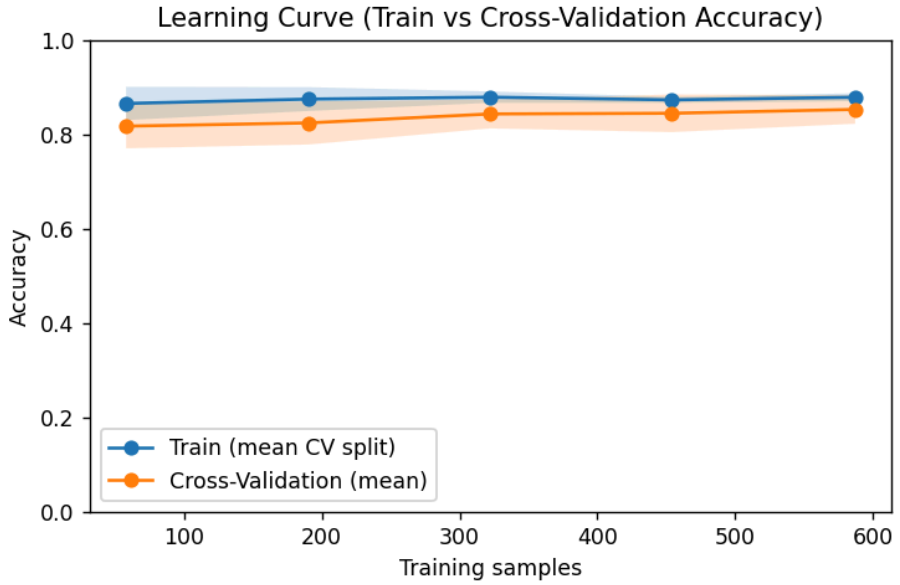



```
Accuracy
  Train: 0.8909657320872274
  Test : 0.8695652173913043

Bias diagnosis: LOW
Explanation: High training accuracy (≥80%). The model captures the relationship between variables well (low bias).
```

Imagen 3: Gráfico de accuracy en train y test, impresión de resultado de bias 2

Diagnóstico y explicación el grado de varianza: bajo medio alto




```
Variance diagnosis: LOW
Generalization gap (Train - Test): 0.0024
Explanation: Train and test accuracies are very close (<5pp difference). The model generalizes well (low variance).
```

Imagen 4: Gráfico de curva de aprendizaje. Junto con resultado impreso en consola.

Para la varianza, también se agregó un poco de código para generar gráficos de línea de aprendizaje, así como una explicación en texto. La varianza del modelo se evaluó comparando el desempeño en el conjunto de entrenamiento y de prueba, así como mediante la curva de aprendizaje generada con validación cruzada. En este caso, la diferencia entre el accuracy de entrenamiento (87.7 %) y el de

prueba (87.5 %) fue de apenas 0.2, que viéndolo desde una perspectiva estadística, fue buen resultado, lo que corresponde a un diagnóstico de varianza baja. La curva de aprendizaje confirma este resultado: las líneas de entrenamiento y validación cruzada se mantienen cercanas entre sí a medida que aumenta el tamaño del conjunto de entrenamiento, sin mostrar una brecha muy grande. Esto indica que el modelo generaliza bien y no depende excesivamente de los datos de entrenamiento, evitando así problemas de overfitting.

Diagnóstico y explicación el nivel de ajuste del modelo: underfitt fitt overfitt

Para determinar si el modelo presenta underfitting, overfitting o un ajuste adecuado, se consideraron todos los puntos anteriores agregados, así como, los que ya estaban presentes en entregas anteriores y la curva de validación:

Bias: El modelo obtuvo un accuracy de entrenamiento de 87.7 %, lo que corresponde a un bias bajo, como pudimos ver anteriormente.

Varianza: La diferencia entre el accuracy de entrenamiento (87.7 %) y prueba (87.5 %) fue de solo 0.2 puntos, como vimos en el análisis anterior dando a entender que existe una varianza baja. La curva de aprendizaje refuerza este resultado, mostrando que las líneas de entrenamiento y validación cruzada se mantienen cercanas.

Curva de validación (max_depth):
La gráfica muestra que tanto el desempeño en entrenamiento como en validación cruzada se mantienen altos y cercanos para distintos valores de max_depth. Podemos observar que el accuracy en entrenamiento no crece mientras el de validación cae lo cual es común que suceda cuando existe overfitting, Tampoco se puede ver que ambos sean bajos como para decir que es underfitting. Esto respalda que el modelo se encuentra en un rango de hiper parámetros correctos con buen ajuste.

Matriz de confusión y métricas por clase:
La matriz de confusión del conjunto de pruebas evidencia una distribución equilibrada de aciertos en ambas clases, con pocos errores de clasificación. El classification report muestra valores de precisión, recall y F1-score cercanos al 0.87–0.89 en ambas clases, lo que confirma que el modelo tiene un rendimiento consistente y balanceado.

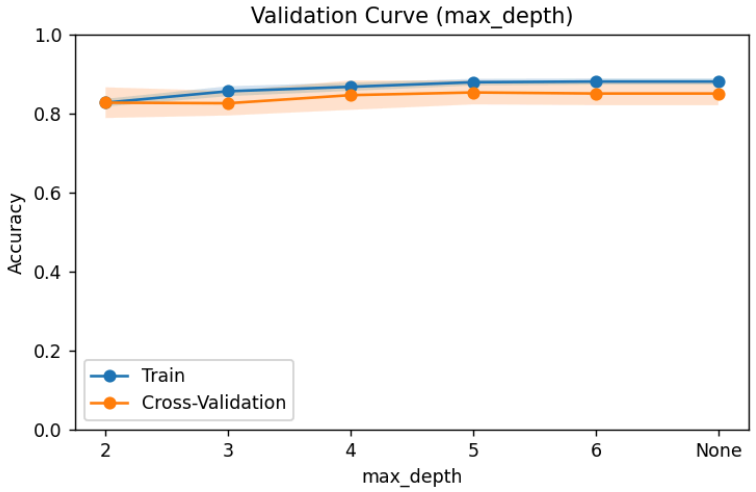


Imagen 5: Gráfico de curva de validación

Per-class Precision, Recall, F1 (Test)

	precision	recall	f1-score	support
0	0.873	0.841	0.857	82
1	0.876	0.902	0.889	102
accuracy			0.875	184
macro avg	0.875	0.872	0.873	184
weighted avg	0.875	0.875	0.875	184

Imagen 6: Classification report obtenido por scikit learn

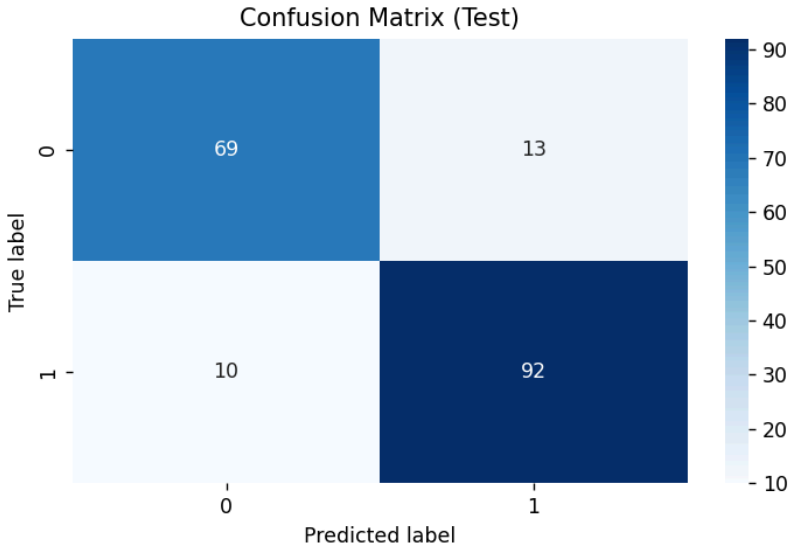


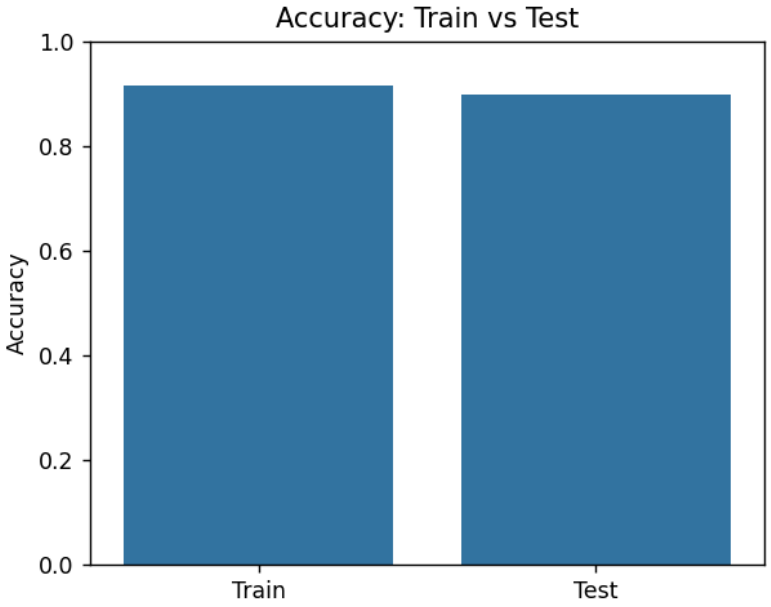
Imagen 7: Matriz de confusión

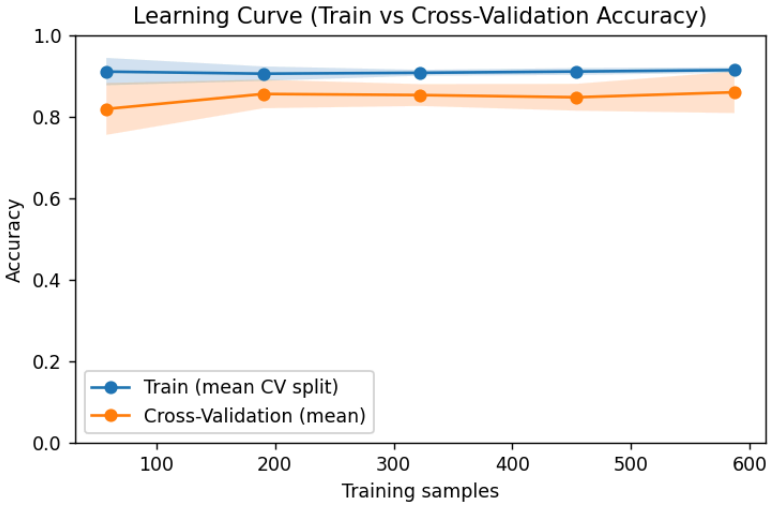
Con base en los resultados anteriores, se concluye que el modelo presenta un ajuste adecuado fit. No se detectan indicios de underfitting ya que el bias es bajo ni de overfitting ya que la varianza es baja y las curvas de validación/aprendizaje no muestran desviaciones significativas. El modelo logra un equilibrio entre aprendizaje de los patrones y capacidad de generalización, ofreciendo un desempeño muy bueno. Pero, trataremos de mejorarlo.

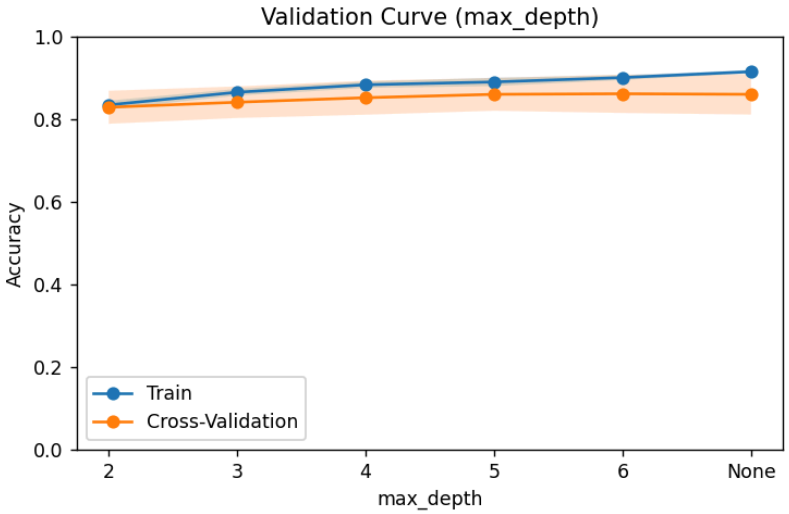
Mejoras después del análisis

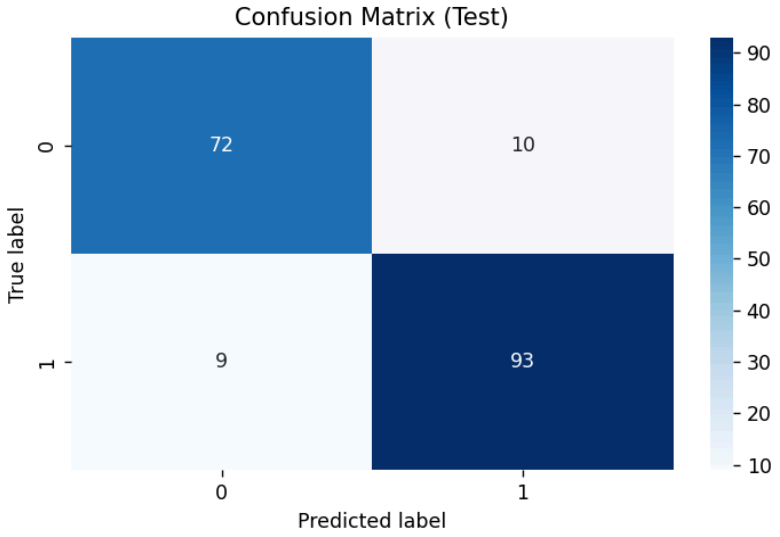
El modelo obtenido, ya era bastante bueno, tenía un buen accuray y también estaba libre de overfitting o bias y varianza altos. Sin embargo, se pudo hacer una ligera mejora en su desempeño general.
Para mejorar esta implementación, se utilizó una técnica llamada bagging que consiste en entrenar varios árboles con diferentes muestras del mismo conjunto de datos y luego combinar sus predicciones. De esta manera, se reduce la variabilidad del modelo y se logra un desempeño un poco más estable y preciso al momento de clasificar nuevos datos, lo que puede llevar a un mejor accuracy y mejores resultados.

Después de hacer este cambio a la implementación se obtuvieron los siguientes resultados:


Accuracy: Train vs Test
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Train
Test


Learning Curve (Train vs Cross-Validation Accuracy)
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Train (mean CV split)
Cross-Validation (mean)
100
200
300
400
500
600
Training samples


Validation Curve (max_depth)
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Train
Cross-Validation
2
3
4
5
6
None
max_depth



Per-class Precision, Recall, F1 (Test)

	precision	recall	f1-score	support
0	0.889	0.878	0.883	82
1	0.903	0.912	0.907	102
accuracy			0.897	184
macro avg	0.896	0.895	0.895	184
weighted avg	0.897	0.897	0.897	184

```
Accuracy
  Train: 0.9141689373297003
  Test : 0.8967391304347826

Bias diagnosis: LOW
Explanation: High training accuracy (≥80%). The model captures relationships well (low bias).

Variance diagnosis: LOW
Generalization gap (Train - Test): 0.0174
Explanation: Train and test accuracies are very close (<5pp difference). The model generalizes well (low variance).
```

A simple vista, es difícil notar la mejora por lo que se hará una tabla comparativa de los resultados:

Métrica	ID3 Anterior	ID3 con bagging
Accuracy en train	0.877	0.914
Accuracy en test	0.875	0.897
Diferencia entre train y test	0.0024	0.017
F1-Score 0	0.857	0.883
F1-Score 1	0.889	0.907
Errores totales en Matriz de confusión	23	19
Bias	Bajo	Bajo
Varianza	Bajo	Bajo
Overfitting ó Underfitting	No	No

Viendo los resultados lado a lado, podemos ver claras mejoras en el modelo nuevo. Sin embargo, podemos notar que la diferencia entre train y test se hizo más grande, no tanto como para significar varianza o bias de un tamaño que pueda afectar la credibilidad, por lo que se considera una mejora total del algoritmo implementado. Es por eso que podemos decir que agregarle bagging fue una buena decisión.

Conclusión

Para finalizar, es importante mencionar que la experiencia de implementar el algoritmo ID3 primero de manera manual y después utilizando un framework permitió comprender de forma más profunda el funcionamiento interno de un modelo de aprendizaje automático. Hacerlo a mano obliga a analizar cada paso del algoritmo, lo que ofrece una visión clara de los fundamentos teóricos. Cuando se hace uso de un framework, se facilita la implementación práctica, ya que proporciona herramientas optimizadas para entrenar, validar y ajustar modelos de forma más eficiente, acercándose a lo que se realiza en entornos profesionales.
Adicionalmente este trabajo también mostró que, aunque un modelo inicial puede dar buenos resultados, se pueden encontrar mejoras, aunque sean muy ligeras, pero eso se puede convertir en una tarea muy complicada, ya que se requiere explorar técnicas adicionales, ajustar hiper parámetros y aplicar métodos como bagging para lograr incrementos como el obtenido, que a pesar de ser pequeño, creo que es valioso. Esto refleja un aspecto importante del aprendizaje automático. Construir un modelo es solo el primer paso, el trabajo real consiste en experimentar, evaluar y perfeccionar constantemente para obtener soluciones más sólidas y confiables que puedan ser utilizadas en situaciones del mundo real.