



Tecnológico de Monterrey

Momento de Retroalimentación: Análisis y Reporte sobre el desempeño del modelo

**Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México**

Eduardo Joel Cortez Valente A01746664

Inteligencia artificial avanzada para la ciencia de datos I

TC3006CB

Grupo: 101

Fecha de entrega:

11 de septiembre de 2023

Índice

Momento de Retroalimentación: Análisis y Reporte sobre el desempeño del modelo	1
Índice	2
Introducción	3
Justificación selección del dataset	3
Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train / Test / Validation)	3
Aplicación del modelo	5
Diagnóstico y explicación	5
Grado de bias o sesgo	7
Grado de varianza	7
Nivel de ajuste del modelo	8
Uso de técnicas de regularización o ajuste de parámetros para mejorar el desempeño	8

Introducción

En el presente reporte se analizará el desempeño del modelo de machine learning diseñado a partir de la librería de Scikit-learn. Dicho modelo hace uso de un árbol de clasificación como algoritmo con el cual analizar los datos que se le han de proporcionar. Es posible observar el código fuente con el cual se hizo el presente reporte en el repositorio donde se encuentra este mismo informe; así con la documentación correspondiente para probar su funcionalidad, entender las variables utilizadas y obtener las gráficas presentadas en el análisis. Igualmente, puede acceder a dicho repositorio haciendo click aquí: [Análisis-desempeño-modelo](#)

Justificación selección del dataset

El dataset utilizado en el presente reporte es Digits. Una de las razones por la cual se eligió dicho dataset es que está ampliamente disponible en bibliotecas y librerías educativas como Scikit-learn, por lo cual es de fácil uso. Adicionalmente, dicho dataset contiene una gran variedad de imágenes de dígitos escritos a mano en diferentes estilos, lo que le da gran diversidad a los datos de entrada; que a su vez permite evaluar de mejor manera la capacidad de lo desarrollado.

Otra cosa a resaltar es que, al hablar de dígitos y ser fácilmente identificables, no hay complejidad en comprender los datos y lo que idealmente deberían representar. Además, la información ha recibido limpieza, ya que tiene un enfoque educativo con propensión a realizar directamente análisis de los modelos.

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

Se realizó la división en conjuntos de entrenamiento, prueba y validación. Dicha división tiene el objetivo de analizar los datos de tal manera que contrastamos el modelo con información que le fue proporcionada e información que no. Del total de los datos, 20% se destina a pruebas y 80% a entrenamiento inicial. Posteriormente, de ese 80% inicial se le vuelve a hacer una división, donde del total 20% se destina a validación y 80% al entrenamiento final.

Al momento de ver los datos a utilizar gráficamente, podemos observar lo siguiente:

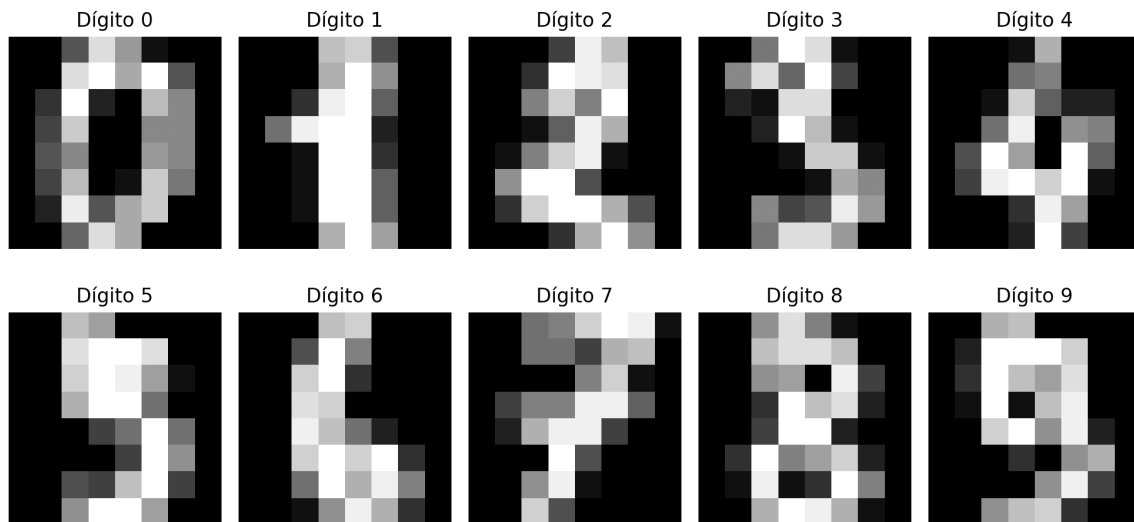


Figura 1. Visualización de los datos con los cuales se trabajará

Para graficar los datos cargados usando `load_digits` de `scikit-learn`, primero reducimos la dimensionalidad de los datos, ya que los datos originales tienen 64 dimensiones (8x8 píxeles) y no se pueden visualizar directamente en un gráfico bidimensional. Una técnica común para reducir la dimensionalidad es el Análisis de Componentes Principales (PCA), técnica que aplique y con la que me es posible ver los siguiente:

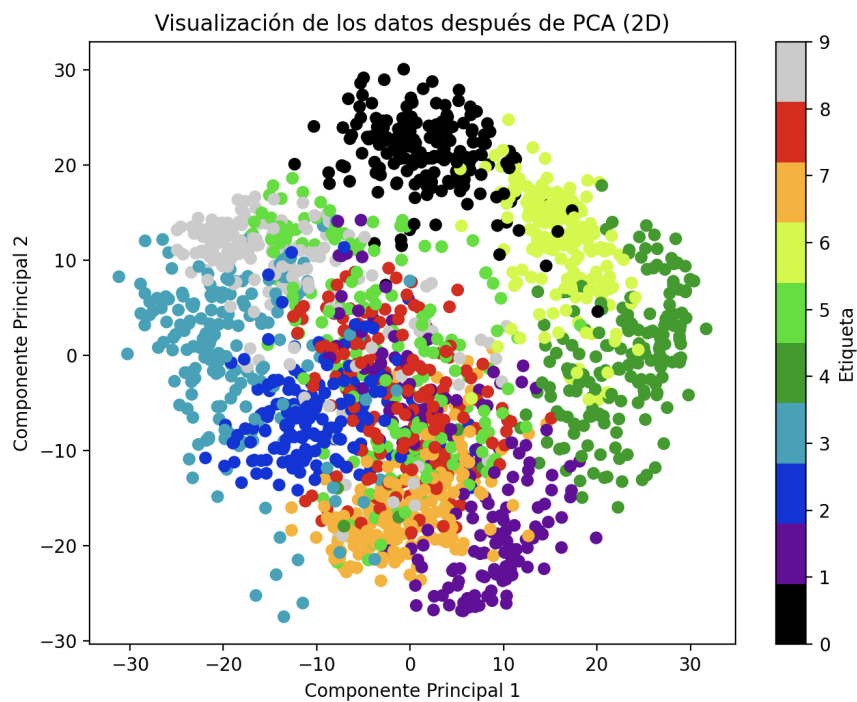


Figura 2. Visualización del dataset después de un tratamiento de PCA para su visualización

Aquí es apreciable visualizar los datos de tal manera que es posible observar por medio de las etiquetas y colores los datos y a que subconjunto idealmente deberían pertenecer.

Aplicación del modelo

Una vez entendido lo anterior, seguiremos con la construcción del modelo y su entrenamiento. Recordando, el algoritmo a utilizar es un árbol de decisión para clasificación; provisto por Scikit-Learn. Se le pasaran los hiperparametros ‘max_depth = 5’, para apreciar el comportamiento del modelo al limitar la probabilidad del árbol según una estimación especulativa inicial.

Diagnóstico y explicación

Después de generar el modelo con esas especificaciones, se realizó el entrenamiento con el conjunto de entrenamiento. El resultado de dicho entrenamiento fue utilizado para predecir su performance con el conjunto de entrenamiento, el conjunto de validación y el conjunto de prueba.

El resultado de las predicciones hechas por dicho modelo se pueden apreciar en la siguientes gráficas:

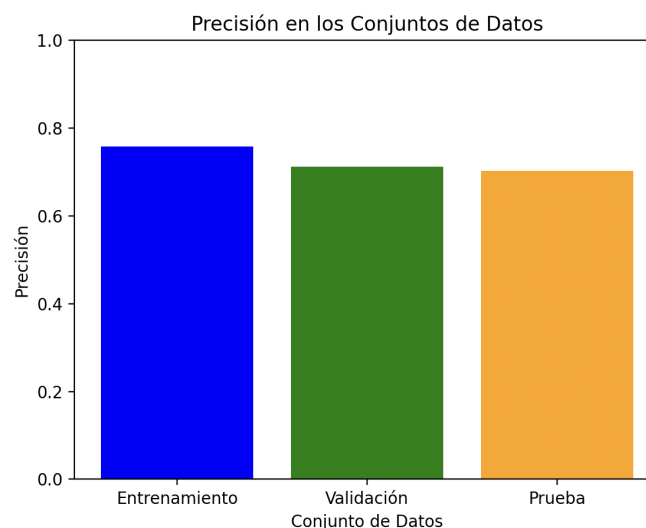


Figura 3. Gráfica de barras comparando la precisión en los conjuntos de datos

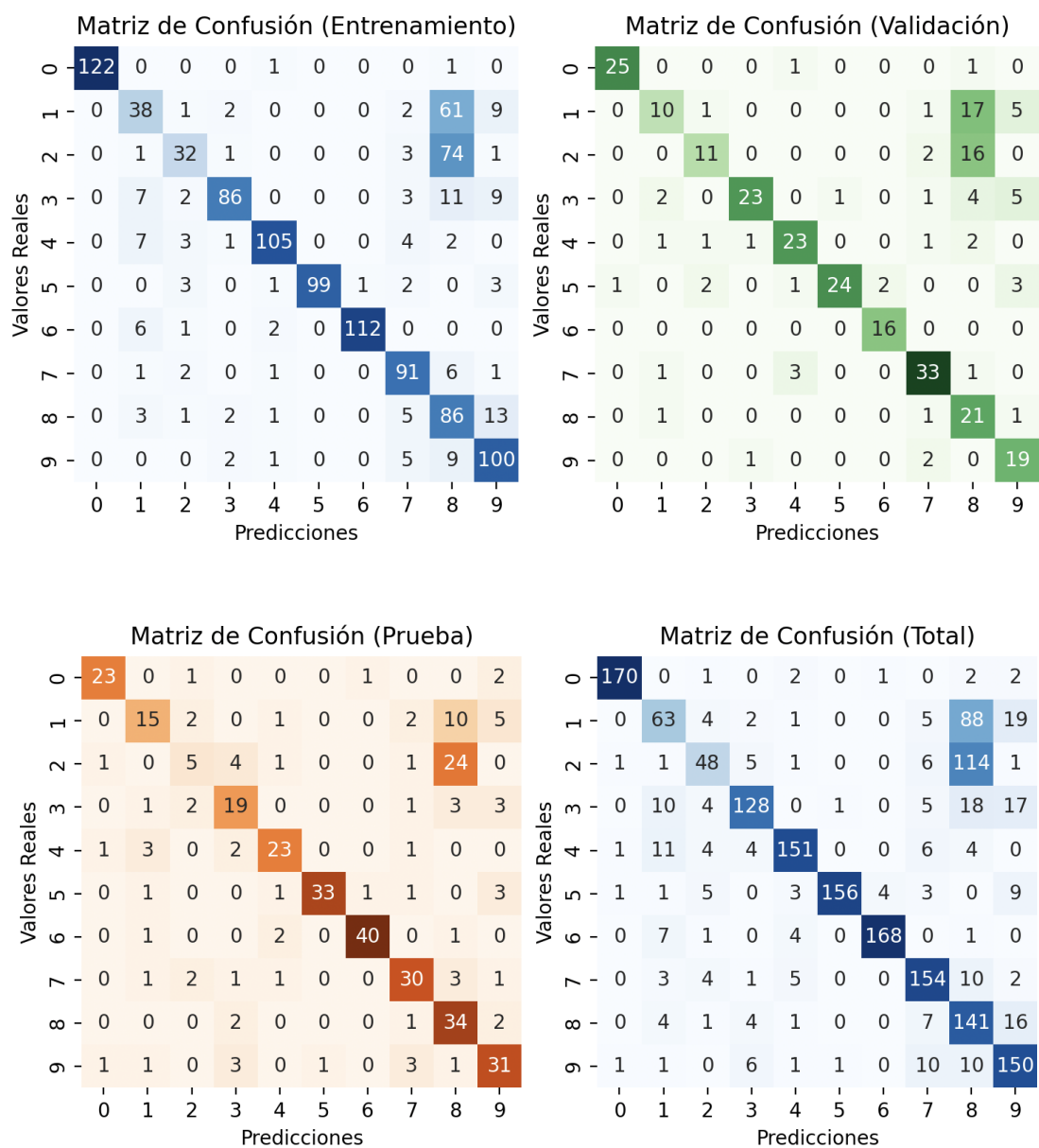


Figura 4. Matrices de confusión graficadas

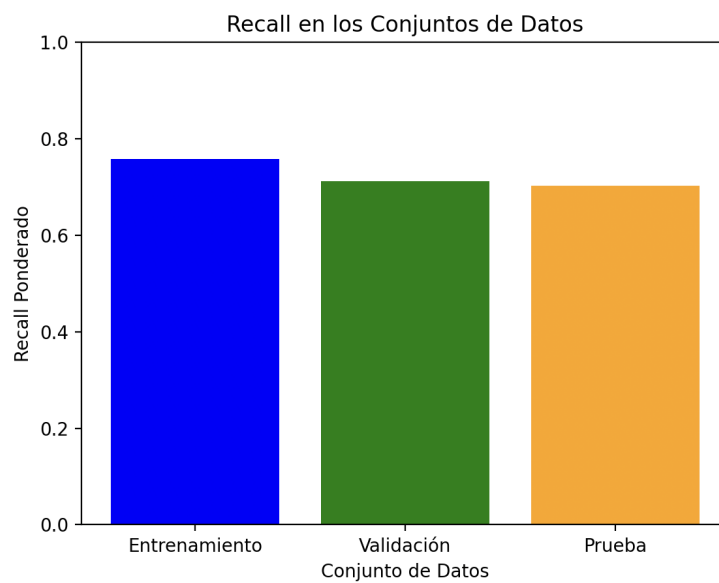


Figura 5. *Gráfica de barras comparando el recall en los conjuntos de datos*

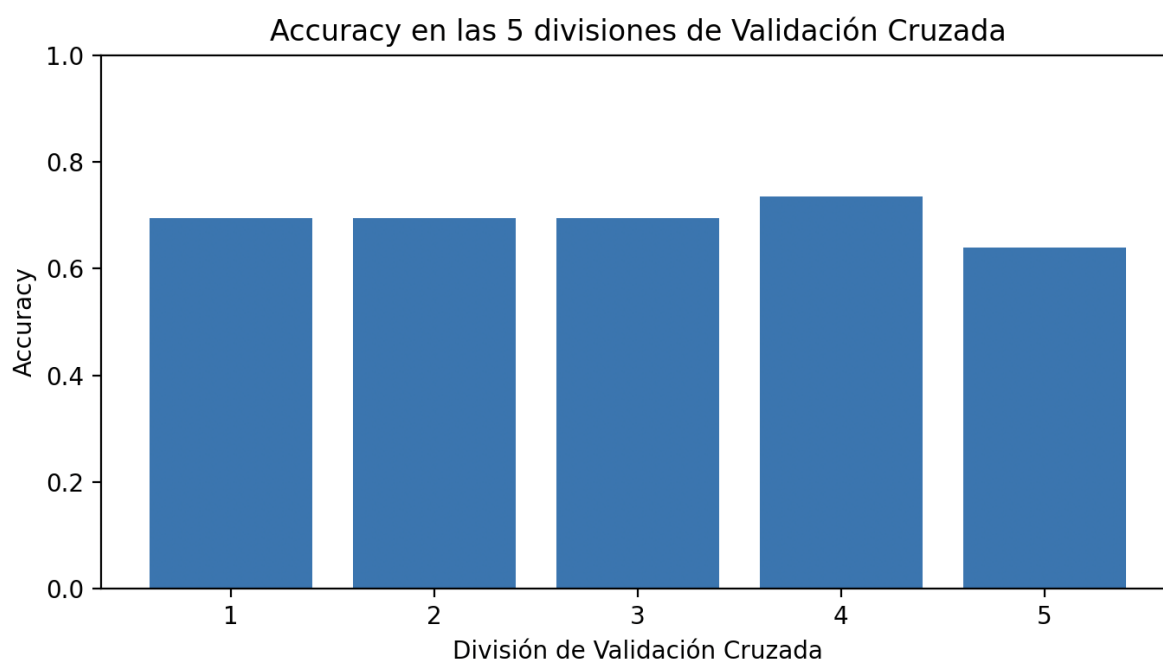


Figura 6. *Gráfica de barras realizando una validación cruzada del conjunto de prueba*

Se realizó una división al conjunto de prueba, donde se aplicó el modelo generado y se realizó un cálculo de su precisión. Dichos resultados nos sirven para apreciar el desempeño del modelo con distintas muestras; y determinar el grado de sesgo y de varianza. Lo que podemos apreciar en la gráfica es que la precisión no es muy alta, pero tiene uniformidad en los resultados que obtiene. Dicha observación se complementa con los siguientes puntos.

Grado de bias o sesgo

De las gráficas anteriores es posible hacer evaluaciones sólidas. Primeramente, las métricas en la precisión y recall de los datos en sus distintas formas sugiere un sesgo bajo. Esto se debe a que los resultados de dichas métricas son consistentes y similares en todos los conjuntos. Podría asumir que el modelo generaliza de manera consistente en diferentes; ya que es capaz de hacerlo de manera consistente en diferentes conjuntos de datos. Por lo cual, la selección del modelo, característica y datos de entrenamiento fueron más que aceptables.

Sin embargo, es importante destacar que las métricas también son uniformemente bajas en todos los conjuntos. Lo que nos inclinaría a considerar que el modelo tiene un sesgo medio. Para cambiar ese comportamiento sería necesario replantear la forma en que se trabaja con los hiperparámetros del modelo, así como explorar otras técnicas de mejora del rendimiento.

Grado de varianza

Revisando las gráficas y comparando los resultados, podemos realizar las siguientes inferencias. Prestando atención a las matrices de confusión, es observable que la del conjunto de validación tiene un patrón similar al del conjunto de entrenamiento. Donde hay una alta cantidad de True Positives y una alta cantidad de False Negatives, se mantiene en todas las matrices de confusión. Sumado a lo previamente mencionado sobre la semejanza de los resultados entre los gráficos de barras de recall y precisión, podemos concluir que el modelo generaliza de manera consistente y tiene una varianza baja.

Nivel de ajuste del modelo

Revisando los datos, es evidente que el rendimiento del conjunto de entrenamiento, validación y prueba son bajos; convergiendo hacia un valor similar a medida que se agregan más datos. Lo anterior hace evidente que el modelo sufre de underfit.

Los criterios de evaluación para concluir que realiza predicciones adecuadas indican que esa aseveración no puede ser verdad. La precisión y el recall señalan el evidente bajo rendimiento de los datos en los tres conjuntos. Especialmente entre el de entrenamiento y validación.

Por otro lado, la matriz de confusión nos indica que los tres conjuntos cuentan con un alto número de falsos negativos en los números ‘uno’ y ‘dos’; al confundirlo con el 8. Eso nos quiere decir que el modelo, debido a los datos y/o hiper-parámetros, no tiene la capacidad de realizar predicciones aceptables; característica que resalta aún más su condición de underfitting.

Uso de técnicas de regularización o ajuste de parámetros para mejorar el desempeño

Posterior al análisis de los resultados obtenidos del análisis del modelo (y con un set de datos especializado para el machine learning); llegue a la conclusión de que los datos, debido al algoritmo de machine learning que seleccioné, no deberían ser una razón de peso para justificar la inexactitud de los resultados. Con ello en mente, determine que el mejor curso de acción que pudo (y pude) haber tomado, es ajustar los hiperparametros.

En este caso, creo que el fallo radica en la profundidad del algoritmo. Le dice una profundidad de cinco, propia para clasificaciones con tres o seis posibles estados. Pero al ser estas imágenes que pueden pertenecer a 10 posibles clases, lo mejor es dar una profundidad de 8. Tampoco excediendo la cantidad de divisiones que haría; puesto que ello implicaría un overfitting.

Otro fallo puede apreciarse en la ausencia de un ‘min_samples_split’. Dicho hiper-parámetro controla el número mínimo de hojas requeridas para dividir un nodo. Es importante evaluar el comportamiento si restringimos desde un primer momento al modelo para asegurarnos de que siga una estructura binaria.

Los resultados el modelo una vez ajustado dicho hiper parámetro es el siguiente:

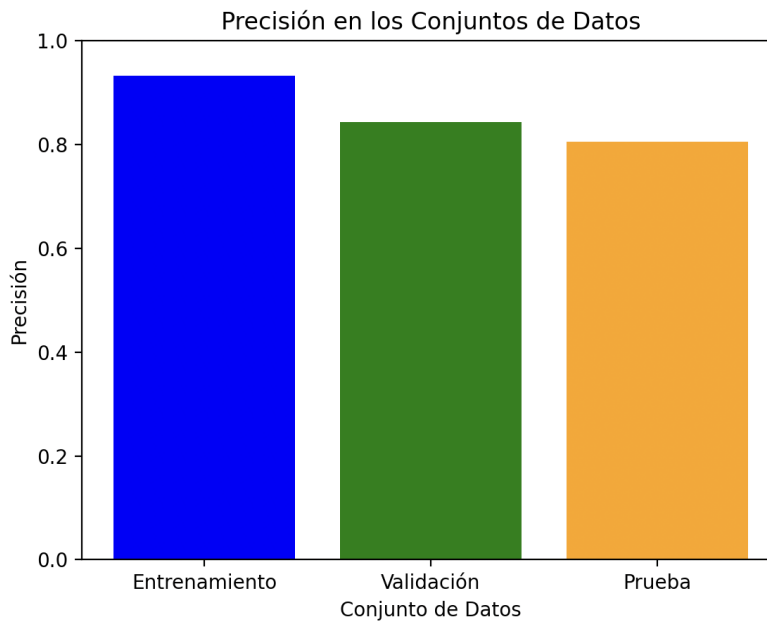


Figura 7. Gráfica de barras comparando la precisión en los conjuntos de datos

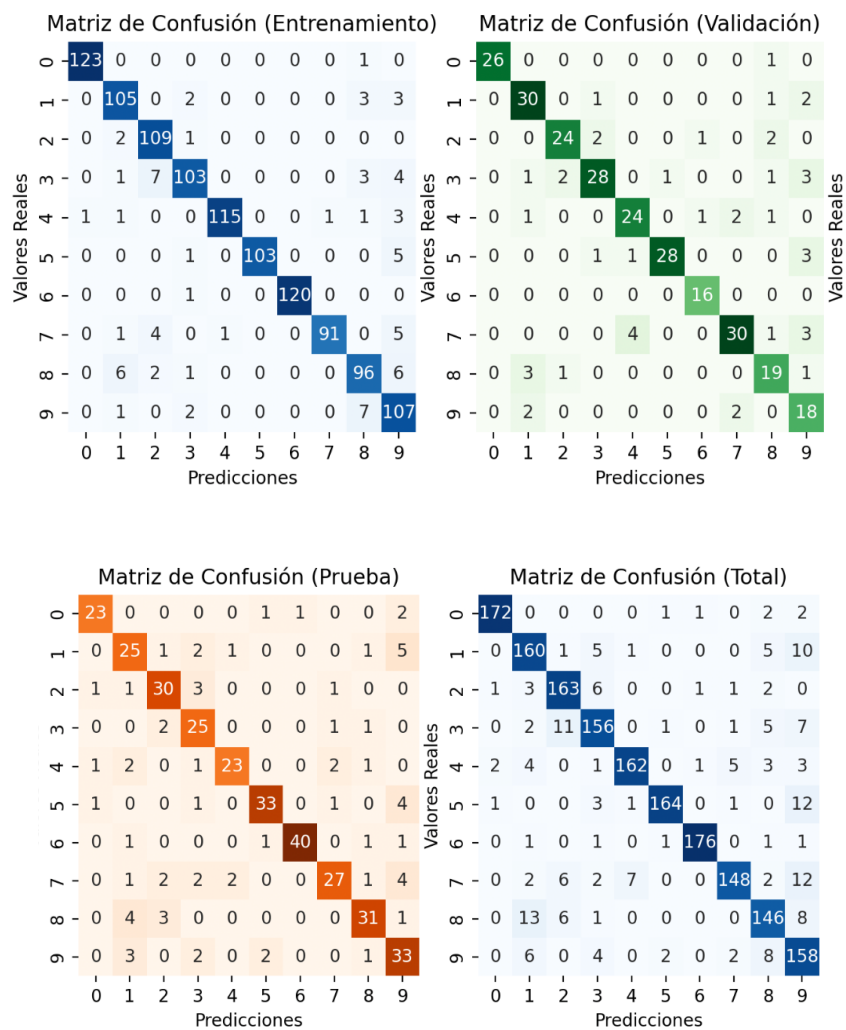


Figura 8. *Matrices de confusión graficadas*

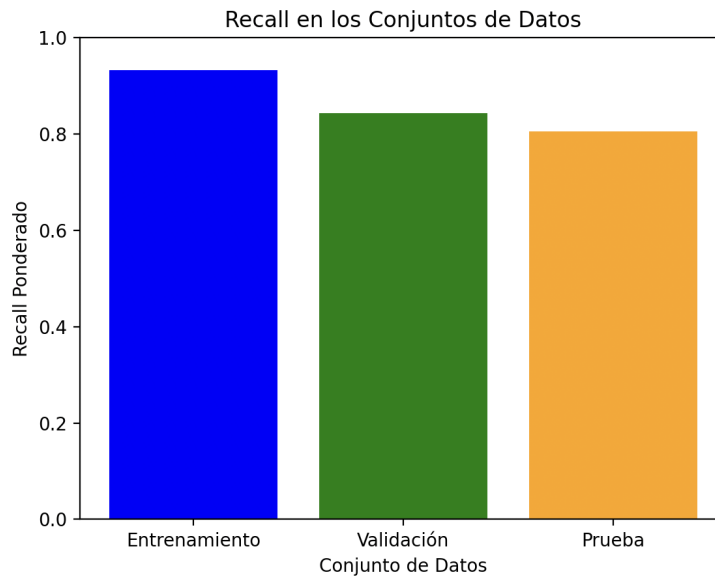


Figura 9. *Gráfica de barras comparando la precisión en los conjuntos de datos.*

Una vez aplicado el ajuste mencionado, podemos apreciar una clara mejora en el modelo generado. Pasamos de estar en un umbral de precisión y recall menor al 80%, a uno que supera el 95% en su conjunto de entrenamiento; cuya validación es mayor al 80% y cuya evaluación en el conjunto de prueba no se queda muy atrás.

Esta notoria mejora no mermó los previos análisis de sesgo y varianza. Al los resultados de las pruebas mantenerse similares, es apreciable que el sesgo es bajo. El problema anterior era que, pese a la uniformidad de los resultados, estos eran muy bajos. Con el ajuste de la profundidad, dicho inconveniente ya no está presente. Por parte de la varianza, las matrices de confusión dejan en evidencia que las confusiones entre el número 'uno' y número 'dos' ya no están presentes en el modelo. Siendo la matriz de confusión de la prueba un ejemplo claro de que la generalización es lo suficientemente acertada como para que el modelo se comporte de una manera esperada al interactuar con datos no vistos previamente.

A lo previamente mencionado, debemos sumarle el hecho de que los resultados de los diferentes conjuntos dejan en evidencia un muy buen rendimiento por parte del modelo. La precisión no es exageradamente alta en los conjuntos de validación y entrenamiento; y su desempeño en el modelo de prueba es más que resaltable. Por lo que me parece pertinente señalar que tiene un ajuste adecuado.