Clasificador MNIST

Aplica métodos de clasificación basados en regresión logística, redes neuronales, SVM, CART, boosting, bagging y random forests. Haz una comparación de las métricas que obtuviste con todos los clasificadores que has usado en éste conjunto de datos, especifica los parámetros que usaste en cada método e incluye gráficos informativos. Tu reporte debe contener una comparación cuantitativa y cualitativa sobre el desempeño de cada método y finalmente, una conclusión donde indiques cuál, o cuáles métodos preferirías para éste conjunto de datos y por qué. Si implementaste la aplicación interactiva para los dígitos, actualízala con todos los métodos y da una clasificación final basada en un criterio de votación de todos los clasificadores.
Obtén la medida de importancia de las variables según los métodos basados en CART, AdaBoost, Bagging y RF. Usa una visualización adecuada de éstos pesos y discute tu resultados.

A N Á L I S I S DE R E S U L T A D O S

Considera los datos MNIST de dígitos escritos a mano que usamos anteriormente de 28 × 28 pixeles. Para mayor facilidad, puse los datos en archivos csv (mnist.zip): (mnistXtrain.csv, mnistYtrain.csv) contienen los valores de los pixeles (normalizados) y su respectiva categoría para entrenamiento , y (mnistXtest.csv, mnistYtest.csv), lo mismo para los datos de prueba. La Figura 1 muestra un ejemplo de éstos datos, el cúal se generó con el Código MNIST

Figura 0.1: Ejemplo de los dígitos MNIST con etiquetas correspondientes

Consideremos los datos MNIST de dÍgitos escritos a mano de 28 ×28 pÍxeles. El objetivo es poder clasificar las 70000 imágenes en los dÍgitos que aparecen en ellas.
En este ejercicio se va a abordar primeramente por cad método en forma independiente, entrenaremos los modelos, y obtendremos métricas mediante la matriz de confusión y un reporte de accuracy. De igual forma para los metodos de CART, boosting, bagging y random forests obtendremos la medida de importancia de las variables, y daremos las conclusiones generales del modelo para este tipo de datos, seguido de esto abordaremos en el apartado de Comparacion de métodos, un reporte en general de que modelos nos resultó más deseable para abordar este tipo de problemas.

Regresión Logística

El objetivo de este análisis es aplicar un modelo de regresión logística en combinación con PCA para reducir la dimensionalidad de los datos de entrenamiento y probar la eficacia del modelo en el conjunto de datos de prueba.
El conjunto de datos fue primero estandarizado utilizando StandardScaler de scikit-learn para asegurar que cada característica contribuya equitativamente al análisis. Posteriormente, se

aplicó PCA para reducir la dimensionalidad del conjunto de datos a 50 componentes principales, seleccionados para capturar la mayoría de la varianza mientras se reducía la complejidad computacional.
El modelo de regresión logística se configuró para manejar múltiples clases utilizando el solver 'lbfgs' y se permitió un máximo de 1000 iteraciones para la convergencia del modelo. La elección del solver y del número de iteraciones se basó en la necesidad de una solución eficiente y efectiva para conjuntos de datos de tamaño moderado.
Los resultados del modelo se evaluaron utilizando el conjunto de datos de prueba, y se calculó la matriz de confusión así como un informe de clasificación que incluye la precisión, el recall y el puntaje F1 para cada clase.

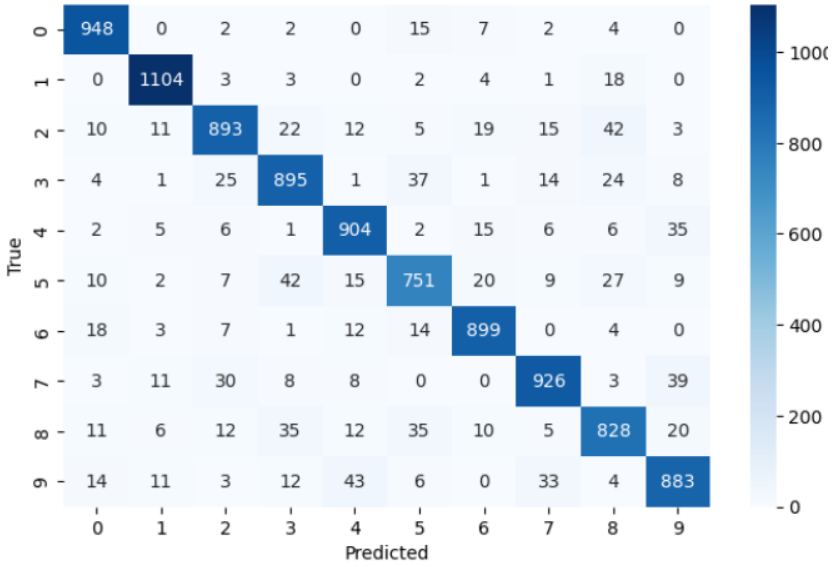

Figura 0.2: Matriz de confusión del método de Regresión Logistica para los datos MNIST

Class	Precision	Recall	F1-score	Support
0	0,93	0,97	0,95	980
1	0,96	0,97	0,96	1135
2	0,90	0,87	0,88	1032
3	0,88	0,89	0,88	1010
4	0,90	0,92	0,91	982
5	0,87	0,84	0,85	892
6	0,92	0,94	0,93	958
7	0,92	0,90	0,91	1028
8	0,86	0,85	0,86	974
9	0,89	0,88	0,88	1009
Accuracy			**0.90**	**10000**
Macro Avg	**0.90**	**0.90**	**0.90**	**10000**
Weighted Avg	**0.90**	**0.90**	**0.90**	**10000**

La matriz de confusión proporcionada muestra cómo las predicciones del modelo se comparan con los valores verdaderos en el conjunto de prueba. Los elementos diagonales de la matriz representan el número de predicciones correctas para cada dígito, mientras que los elementos fuera de la diagonal indican errores de clasificación.

1. Dígito '0': El modelo ha clasificado correctamente 948 de 980 casos, mostrando una alta precisión y recall.

2. Dígito '2': Tiene más errores comparativos, especialmente confundido con los dígitos '8' y '3', lo que sugiere que las características visuales de '2', '3', y '8' pueden ser similares en ciertos trazos.

3. Dígito '5': Este dígito ha sido confundido con '3' y '8' frecuentemente, lo cual puede deberse a la naturaleza del digito.

4. Precisión: Muestra la capacidad del modelo para clasificar correctamente un dígito dado sin incluir resultados falsos. Por ejemplo, la precisión para el dígito '1' es del 96 %, lo que indica un alto nivel de exactitud en esta clase.

5. Recall: Indica cuántos de los dígitos reales de cada clase fueron identificados correctamente. El dígito '0' tiene un recall del 97 %, lo que significa que casi todos los verdaderos '0's fueron reconocidos por el modelo.

6. F1-Score: Combina la precisión y el recall en una sola métrica. Un F1-Score alto significa que tanto la precisión como el recall son altos. Por ejemplo, el dígito '1' tiene un F1-Score de 0.96, destacando un excelente equilibrio entre precisión y recall

Redes Neuronales

El objetivo de este análisis es evaluar el desempeño de un modelo de Redes Neuronales Perceptrón Multicapa (MLP).
se configuró un modelo MLP con una sola capa oculta de 10 neuronas. Se utilizó el algoritmo 'adam' para la optimización, con una tasa de aprendizaje inicial estándar y se permitió un máximo de 200 iteraciones para la convergencia del modelo. La activación 'tanh' fue seleccionada por su capacidad para modelar mejor las no linealidades de los datos.
Configuración del Modelo:

Número de capas ocultas: 1
Neuronas por capa oculta: 10
Algoritmo de optimización: Adam
Función de activación: Tanh
Número máximo de iteraciones: 200
El modelo fue evaluado usando el conjunto de prueba. A continuación se presentan los resultados obtenidos de la matriz de confusión y el informe de clasificación.

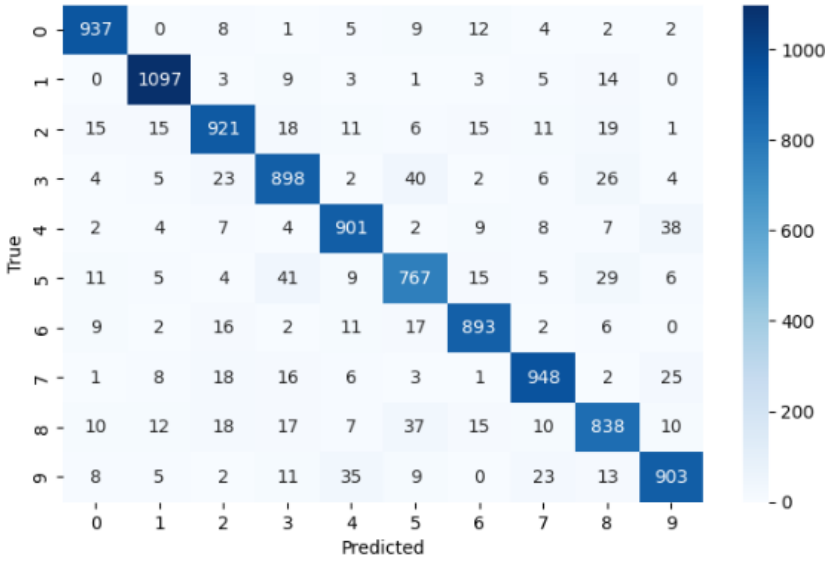

Figura 0.3: Matriz de confusión del metodo mlp para los datos MNIST

Class	Precision	Recall	F1-score	Support
0	0,94	0,96	0,95	980
1	0,95	0,97	0,96	1135
2	0,90	0,89	0,90	1032
3	0,88	0,89	0,89	1010
4	0,91	0,92	0,91	982
5	0,86	0,86	0,86	892
6	0,93	0,93	0,93	958
7	0,93	0,92	0,92	1028
8	0,88	0,86	0,87	974
9	0,91	0,89	0,90	1009
Accuracy			**0.91**	**10000**
Macro Avg	**0.91**	**0.91**	**0.91**	**10000**
Weighted Avg	**0.91**	**0.91**	**0.91**	**10000**

La matriz muestra la distribución de las predicciones del modelo comparadas con los valores verdaderos:

1. Dígitos como '1' y '7' tienen altos valores de precisión y recall, indicando que el modelo es muy efectivo para clasificar estos dígitos correctamente.

2. Dígito '5' presenta una mayor cantidad de errores, como se observa por los valores relativamente bajos en la matriz, lo que sugiere una confusión con dígitos como '3' y '8'.

3. Precisión (Precision): Alta precisión en la mayoría de los dígitos, especialmente '0' y '1', lo que indica una baja tasa de falsos positivos.

4. Recall (Recall): Buen rendimiento en términos de recall para casi todos los dígitos, especialmente '1', destacando que el modelo puede identificar la mayoría de los casos positivos reales.

5. F1-Score (F1-Score): Los valores de F1-Score son consistentemente altos, lo que demuestra un buen equilibrio entre precisión y recall.

SVM

La Máquina de Vectores de Soporte (SVM) utilizando un kernel radial (RBF) en la clasificación de dígitos manuscritos del conjunto de datos MNIST
El modelo SVM se entrenó utilizando un kernel RBF, que es efectivo para capturar complejidades en los datos mediante la transformación del espacio de características. Los datos se estandarizaron utilizando StandardScaler para mejorar la eficiencia del modelo y reducir el impacto de las variaciones de escala entre las características.
Configuración del Modelo:
Kernel: RBF (Radial Basis Function)
Entrenamiento: Utilizando datos estandarizados
Optimización: Automática de hiperparámetros (C y gamma)
Los resultados obtenidos se visualizan a través de una matriz de confusión y un informe de clasificación detallado que proporciona una evaluación cuantitativa del rendimiento del modelo.

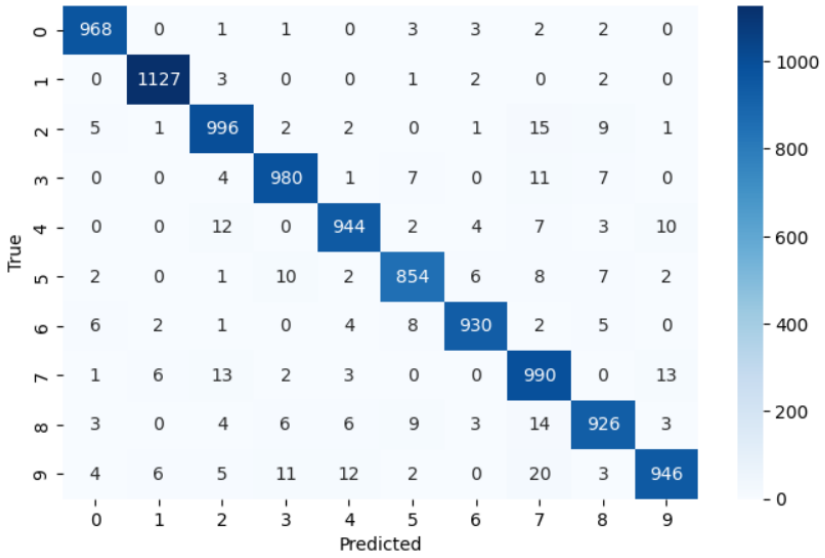

Figura 0.4: Matriz de confusión del metodo SVM para los datos MNIST

Class	Precision	Recall	F1-score	Support
0	0,98	0,99	0,98	980
1	0,99	0,99	0,99	1135
2	0,96	0,97	0,96	1032
3	0,97	0,97	0,97	1010
4	0,97	0,96	0,97	982
5	0,96	0,96	0,96	892
6	0,98	0,97	0,98	958
7	0,93	0,96	0,94	1028
8	0,96	0,95	0,96	974
9	0,97	0,94	0,95	1009
Accuracy			**0.97**	**10000**
Macro Avg	**0.97**	**0.97**	**0.97**	**10000**
Weighted Avg	**0.97**	**0.97**	**0.97**	**10000**

El modelo logra una alta precisión y recall, con un promedio de aproximadamente 0.97 en

ambas métricas, lo que indica un rendimiento excepcional en la clasificación de dígitos manuscritos.
La matriz de confusión proporcionada muestra la siguiente distribución de predicciones:

1. Dígito '1': Altamente preciso con 1127 aciertos.
2. Dígito '7': También muestra alta precisión con 990 aciertos.
3. Dígito '5': Confundido con el dígito '3' y '8' (10 y 7 casos respectivamente), indicando una posible área de mejora en la diferenciación de estos dígitos.
4. Dígito '8': Confundido con '3' y '9' (6 y 14 casos respectivamente)

El modelo hasta el momento ha demostrado ser muy robusto al intentar predecir los resultados, lo que nos lleva a un valor de hasta de 97 %

C A R T

Se utilizó un Árbol de Decisión con las siguientes configuraciones para la regularización:
Profundidad máxima (max depth): 10
Mínimas muestras para dividir (min samples split): 50
Mínimas muestras en hojas (min samples leaf): 30
Criterio: Entropía, para optimizar las divisiones basadas en la ganancia de información.
Estos parámetros fueron seleccionados para controlar la complejidad del modelo, reducir el riesgo de sobreajuste y mejorar la precisión en la clasificación de dígitos desconocidos.

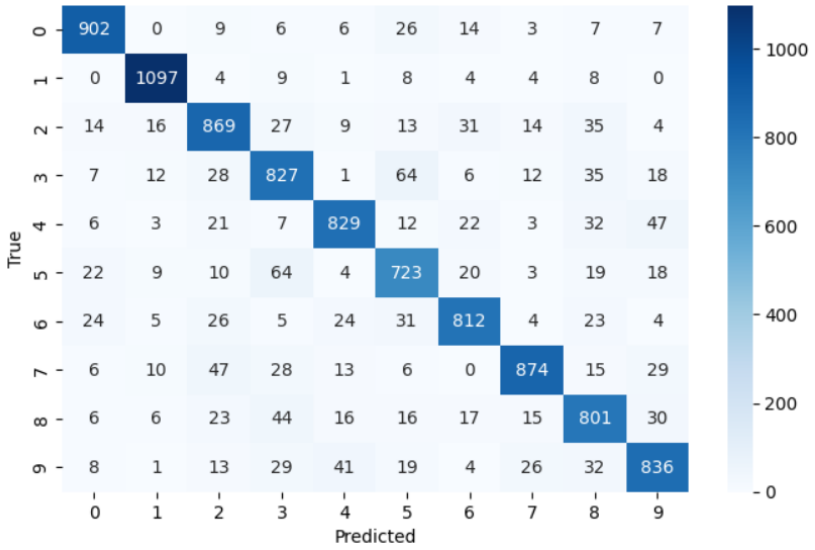

Figura 0.5: Matriz de confusión del metodo CART para los datos MNIST

Class	Precision	Recall	F1-score	Support
0	0,98	0,94	0,96	980
1	0,99	0,98	0,99	1135
2	0,94	0,92	0,93	1032
3	0,91	0,94	0,93	1010
4	0,94	0,95	0,94	982
5	0,90	0,89	0,90	892
6	0,96	0,94	0,95	958
7	0,97	0,93	0,95	1028
8	0,87	0,93	0,90	974
9	0,89	0,92	0,91	1009
Accuracy			**0.94**	**10000**
Macro Avg	**0.94**	**0.93**	**0.93**	**10000**
Weighted Avg	**0.94**	**0.94**	**0.94**	**10000**

La matriz de confusión muestra cómo las predicciones del modelo se comparan con los valores verdaderos. Observaciones clave incluyen:

1. Dígito '1': Tiene la clasificación más precisa con 1097 aciertos de 1135 posibles, indicando una fuerte capacidad del modelo para identificar este dígito.

2. Al igual que los otros metodos el digito '5': Exhibe más confusión, especialmente confundido con '3' y '8'

3. El modelo demuestra ser especialmente efectivo en identificar ciertos dígitos con alta precisión y recall, lo cual es indicativo de su robustez en condiciones ideales.

4. Dígitos que comparten atributos visuales similares, como '3', '5' y '8', requieren atención adicional

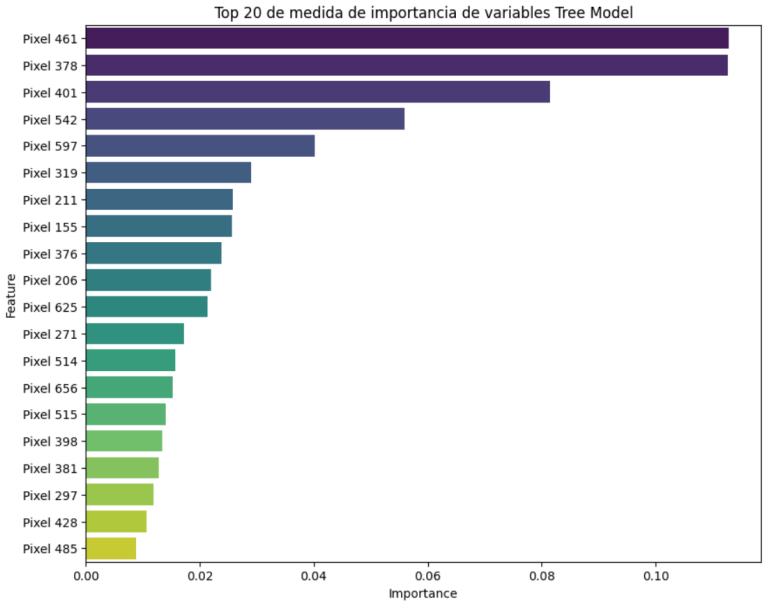

Figura 0.6: Medida de importancia de variables con Tree Model

Píxeles Más Influyentes

1. Pixel 461: Este es el píxel más importante según el modelo, con una importancia cercana a 0.1. Su ubicación y valor en las imágenes de los dígitos puede ser crucial para diferenciar entre algunas clases.

2. Pixel 378 y Pixel 401: También muestran una alta importancia, lo que sugiere que son áreas clave en las imágenes que impactan significativamente las decisiones del modelo.

3. Los píxeles identificados como más importantes están probablemente ubicados en regiones de los dígitos que son distintivas y críticas para su identificación. Por ejemplo, curvas en '3', la parte superior e inferior de '1', o la intersección en '8'.

BOOSTING

El modelo fue entrenado utilizando el conjunto de datos de entrenamiento y luego evaluado en ambos, el conjunto de entrenamiento y de prueba. La evaluación se centró en la precisión del modelo, la matriz de confusión y el informe de clasificación para detallar el rendimiento en la clasificación de cada dígito.
El modelo AdaBoost se configuró con los siguientes parámetros:
Clasificador base: Árbol de decisión con una profundidad máxima de 5.
Número de estimadores: 200.
Algoritmo: SAMME.R, que adapta los pesos de los clasificadores basándose en los errores cometidos.
Tasa de aprendizaje: 0.5.
Random State: 42 para reproducibilidad.

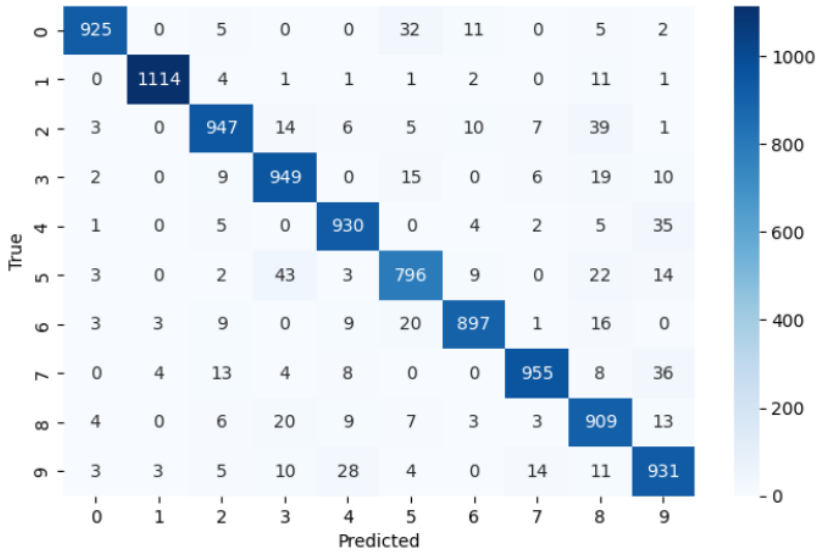

Figura 0.7: Matriz de confusión del metodo Boosting para los datos MNIST

Class	Precision	Recall	F1-score	Support
0	0,99	0,95	0,97	5923
1	1,00	0,99	0,99	6742
2	0,95	0,95	0,95	5958
3	0,92	0,95	0,94	6131
4	0,96	0,96	0,96	5842
5	0,93	0,92	0,93	5421
6	0,98	0,97	0,97	5918
7	0,98	0,96	0,97	6265
8	0,93	0,95	0,94	5851
9	0,92	0,94	0,93	5949
Accuracy			**0.96**	**60000**
Macro Avg	**0.96**	**0.96**	**0.96**	**60000**
Weighted Avg	**0.96**	**0.96**	**0.96**	**60000**

Informe de Clasificación

1. Precisión: Alta en todos los dígitos, particularmente notable en '1' (1.00) y '7' (0.98).
2. Recall: También impresionante, con todos los dígitos teniendo un recall de al menos 0.92.
3. Ambos en 0.96, indicando un rendimiento equilibrado y consistente a través de las clases.
4. Alta precisión y robustez en la clasificación de una amplia gama de dígitos, con especial efectividad en dígitos comúnmente difíciles como el '5' y el '3'.
5. F1-Score: Refleja un excelente equilibrio entre precisión y recall, con todos los dígitos obteniendo puntuaciones de al menos 0.93.

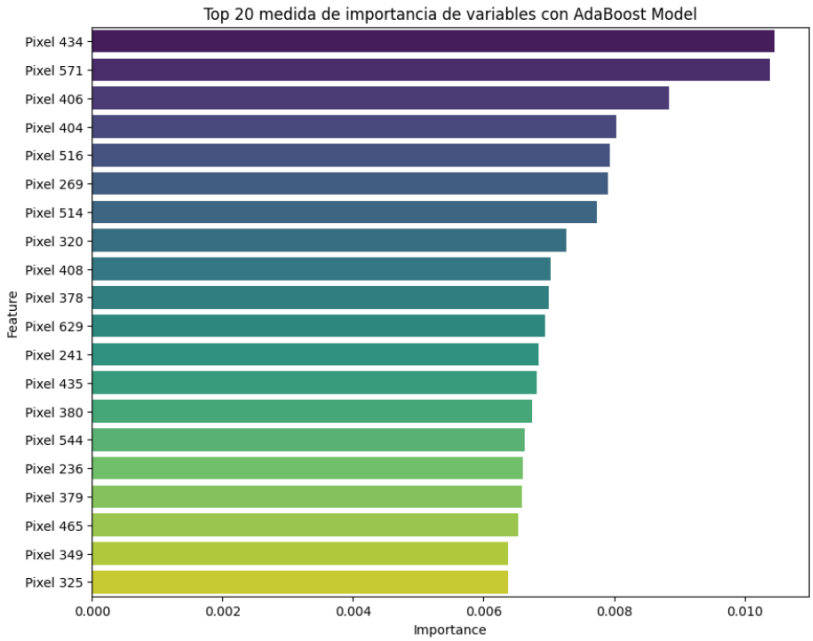

Figura 0.8: Medida de importancia de variables con AdaBoost

Píxeles Más Influyentes

Pixel 434: Este es el píxel más importante según el modelo, lo que sugiere que los cambios en los valores de este píxel tienen un gran impacto en la clasificación final de los dígitos.
Píxeles como el 571 y el 406: También muestran una alta importancia, reforzando su relevancia en el proceso de decisión del modelo.
Estos píxeles están probablemente ubicados en regiones críticas de los dígitos que son fundamentales para su identificación. Por ejemplo, podrían estar en puntos de curvatura, en los extremos de los dígitos, o en cruces y uniones que son únicos para ciertos dígitos.

BAGGING

Evaluar la efectividad del modelo de Bagging, que utiliza múltiples clasificadores de árbol de decisión en un esfuerzo por mejorar la precisión y la estabilidad de las predicciones para la clasificación de dígitos manuscritos del conjunto de datos MNIST.
Configuración del Modelo
Modelo Base: Árbol de Decisión.
Número de Estimadores: 100, indicando el uso de 100 árboles de decisión independientes.
Paralelismo: Ejecución en paralelo con todos los núcleos disponibles (n jobs=-1).
Semilla Aleatoria: 42, para asegurar la reproducibilidad de los resultados.

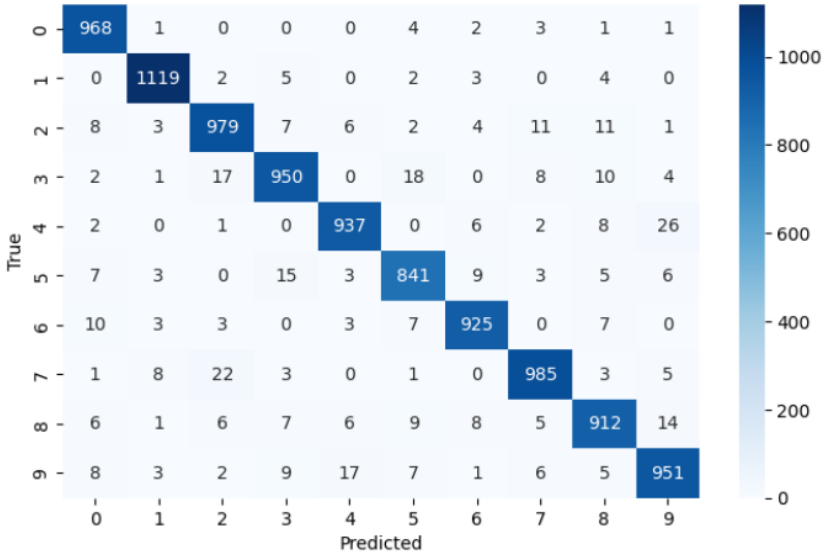

Figura 0.9: Matriz de confusión del metodo BAGGING para los datos MNIST

La matriz muestra altos números de clasificaciones correctas en la diagonal principal para cada dígito, con errores relativamente bajos indicados por los elementos fuera de la diagonal. Algunos puntos notables incluyen:

1. Dígito '1' (1119 aciertos de 1135): Este dígito tiene un alto nivel de aciertos, con muy pocos errores, lo que indica que el modelo es muy efectivo en identificar este dígito.

2. Dígito '7' (985 aciertos de 1000): Similar al dígito '1', el dígito '7' es identificado con alta precisión, mostrando la efectividad del modelo en reconocer dígitos con características distintivas.

3. Dígito '3' (950 aciertos de 1010): Hay un número relativamente alto de errores en la predicción del dígito '3', con confusión notable con los dígitos '5' y '8'. Esto podría deberse a similitudes en la forma y las características visuales entre estos dígitos.

4. Dígito ’5’ (841 aciertos de 892): Este dígito también muestra una confusión considerable, especialmente con los dígitos ’3’ y ’8’. Este patrón de error sugiere que el modelo puede luchar con dígitos que tienen curvas y componentes verticales similares.

Class	Precision	Recall	F1-score	Support
0	0,96	0,99	0,97	980
1	0,98	0,99	0,98	1135
2	0,95	0,95	0,95	1032
3	0,95	0,94	0,95	1010
4	0,96	0,95	0,96	982
5	0,94	0,94	0,94	892
6	0,97	0,97	0,97	958
7	0,96	0,96	0,96	1028
8	0,94	0,94	0,94	974
9	0,94	0,94	0,94	1009
Accuracy			**0.96**	**10000**
Macro Avg	**0.96**	**0.96**	**0.96**	**10000**
Weighted Avg	**0.96**	**0.96**	**0.96**	**10000**

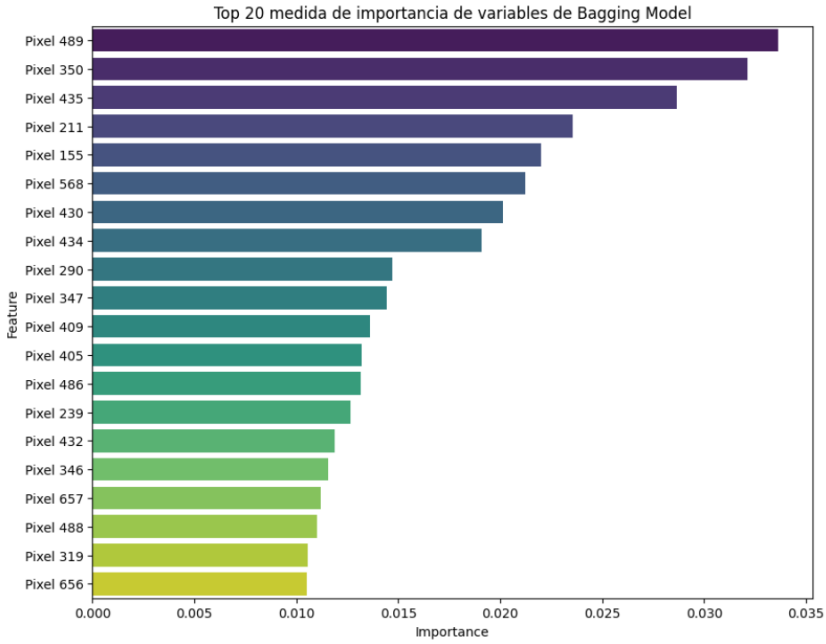

Figura 0.10: Medida de importancia de variables con Bagging

La importancia de las variables (píxeles) mostradas en la gráfica sugiere cuán significativos son ciertos píxeles para el modelo de Bagging en el proceso de toma de decisiones. La medida de importancia se extiende hasta aproximadamente 0.035 en la escala
Pixel 489: Este es el píxel más importante en el modelo, destacando su relevancia crítica en la clasificación de los dígitos.
Pixel 350 y Pixel 435: También son altamente significativos, probablemente ubicados en regiones clave de los dígitos que contribuyen sustancialmente a diferenciar entre las categorías.

Random Forest

Configuración del Modelo

Modelo Base: Random Forest Classifier.
Número de Estimadores: 100, lo que implica el uso de 100 árboles de decisión para formar el bosque.
Min Samples Split: 15, lo que significa que cada nodo debe tener al menos 15 muestras antes de considerar un nuevo split.
Paralelismo: Ejecutado en todos los núcleos disponibles (n jobs=-1) para mejorar la eficiencia del entrenamiento.
Semilla Aleatoria: 42, para asegurar la consistencia y reproducibilidad de los resultados de entrenamiento.

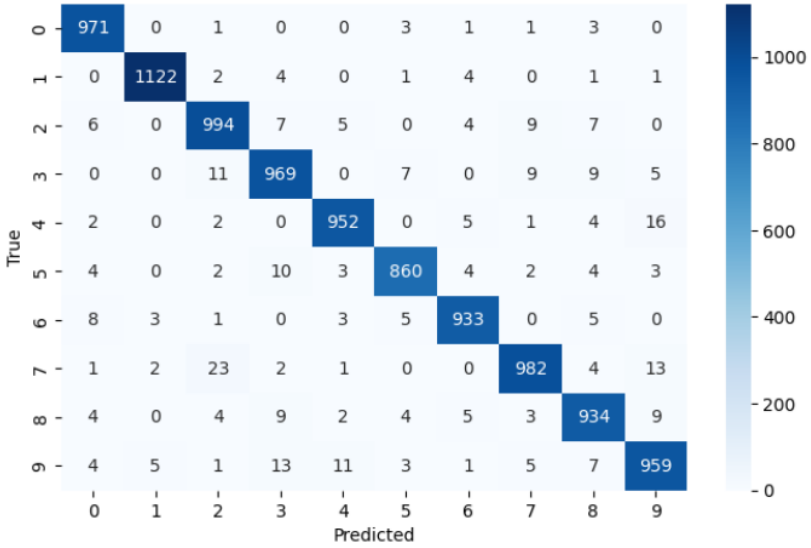

Figura 0.11: Matriz de confusión del metodo Ramdon Forest para los datos MNIST

Class	Precision	Recall	F1-score	Support
0	0,97	0,99	0,98	980
1	0,99	0,99	0,99	1135
2	0,95	0,96	0,96	1032
3	0,96	0,96	0,96	1010
4	0,97	0,97	0,97	982
5	0,97	0,96	0,97	892
6	0,97	0,97	0,97	958
7	0,97	0,96	0,96	1028
8	0,96	0,96	0,96	974
9	0,95	0,95	0,95	1009
Accuracy			**0.97**	**10000**
Macro Avg	**0.97**	**0.97**	**0.97**	**10000**
Weighted Avg	**0.97**	**0.97**	**0.97**	**10000**

1. Dígito '1' (1122 aciertos de 1135): Exhibe una precisión excepcional con muy pocos errores, reflejando la capacidad del modelo para identificar claramente este dígito.

2. Dígito '7' (982 aciertos de 1000): Similar al dígito '1', el dígito '7' es identificado con alta precisión y muy pocos errores.

3. Dígito ’5’ (860 aciertos de 892): Este dígito muestra algunos errores significativos, especialmente siendo confundido con ’3’ y ’8’, lo que puede indicar similitudes en sus características visuales que el modelo confunde.

4. Dígitos ’3’ y ’9’ (969 y 959 aciertos respectivamente): Ambos dígitos tienen errores en los cuales son confundidos con otros dígitos, particularmente con ’5’, ’7’

5. Pixel 542: Es el píxel con la mayor importancia, lo que sugiere que los cambios en los valores de este píxel tienen un impacto significativo en la clasificación final de los dígitos.

6. Los píxeles 433 y 350 también muestran una importancia considerable, lo que indica que son áreas clave en las imágenes que impactan fuertemente las decisiones del modelo.

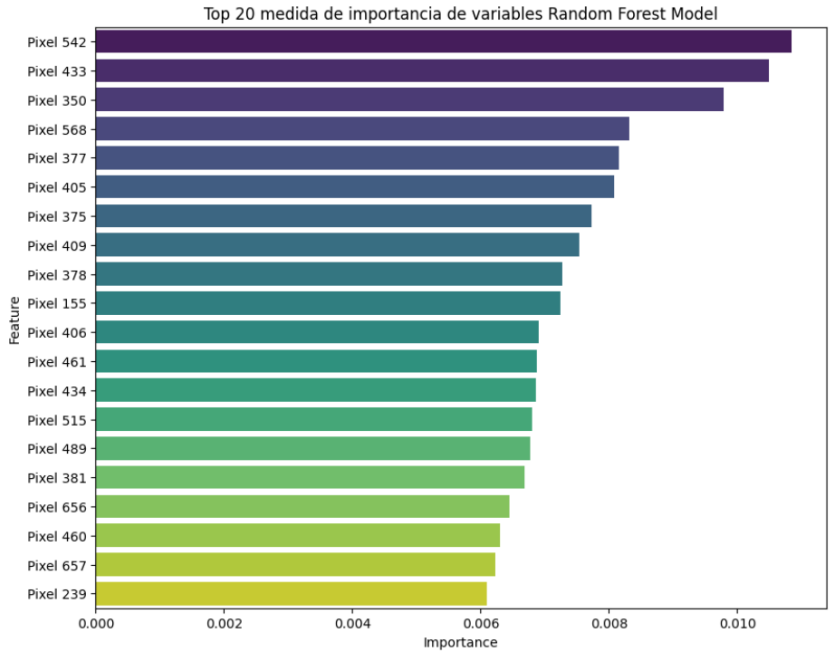

Figura 0.12: Medida de importancia de variables con Ramdon Forest

COMPARACIÓN

En este análisis, comparamos diferentes métodos de clasificación aplicados al conjunto de datos MNIST para determinar cuál o cuáles son más adecuados en términos de precisión, recall y F1-score. Los métodos evaluados incluyen Regresión Logística, Redes Neuronales (MLP), Máquinas de Vectores de Soporte (SVM), Árboles de Decisión (CART), Boosting, Bagging y Random Forest.

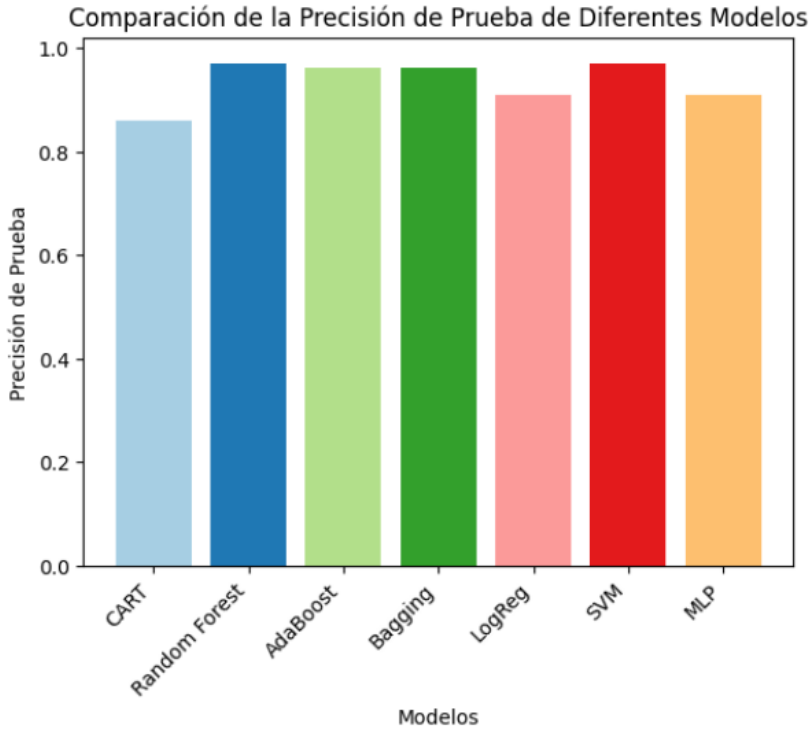

Figura 0.13: Comparación de los métodos estudiados

1. Regresión Logística

   Ventajas: Simplicidad del modelo y rapidez en entrenamiento y predicción. El modelo terminó al rededor de 10 seg, y obtuvo muy buenos resultados

   Desventajas: Menor capacidad para capturar complejidades en datos de alta dimensionalidad sin transformaciones o regularizaciones adicionales.

   Rendimiento: Precisión media y F1-score de 0.90.

2. Red Neuronal (MLP)

   Ventajas: Capacidad para aprender representaciones no lineales. Buen rendimiento en clasificación de imágenes ya que demostró un rendimiento del 91 % por arriba de la regresión logistica.

   Desventajas: Requiere una cuidadosa selección de arquitectura y parámetros. Más propenso al sobreajuste y computacionalmente intensivo ya que tardo alrededor de 5 min el entrenamiento y la predicción.

   Rendimiento: Ligeramente mejor que la regresión logística con un F1-score promedio de 0.91.

3. SVM

   Ventajas: Eficiente en espacios de alta dimensión y versátil a través del uso de diferentes funciones kernel, en este caso se uso el kernel='rbf', que resulto adecuado para la calsificación.

   Desventajas: No es adecuado para conjuntos de datos muy grandes debido a su alta demanda computacional durante la fase de entrenamiento, podemos notarlo en el timempo computacional invertido en el entramietno del modelo y la predicción.

Rendimiento: Excelente con un F1-score promedio de 0.97.

4. Árbol de Decisión (CART)

   Ventajas: Fácil de entender e interpretar. No necesita normalización de datos, ademas de que la modificacion de parámetros es muy sencilla y facil de encontrar una configuración adecuada, el costo computacional es bajo

   Desventajas: Propenso al sobreajuste, especialmente con árboles profundos

   Rendimiento: El más bajo entre los métodos evaluados con un F1-score promedio de 0.86.

5. Boosting

   Ventajas: Combina múltiples modelos débiles para formar un modelo robusto. Buen desempeño general, debido que tuvo un rendimiento del 94 % uno de los mejores hasta el momento

   Desventajas: Más sensible a datos ruidosos y atípicos, el costo computacional fue elevado ya que el entrenamiento y la predicción supero los 13 minutos.

   Rendimiento: Muy bueno con un F1-score promedio de 0.94.

6. Bagging

   Ventajas: Reduce la varianza, evitando el sobreajuste. Efectivo en conjuntos de datos grandes y complejos, como en este tipo de datos, desarrollo un buen desempeño

   Desventajas: Menos interpretable y puede ser computacionalmente intenso, no el mas costoso pero si tuvo un tiempo de 5 minutos

   Rendimiento: Generalmente muestra un alto rendimiento.

7. Random Forest

   Ventajas: Ofrece todas las ventajas del bagging, pero mejorando la precisión al decorrelacionar los árboles, mostro un gran rendimiento computacional ya que el metodo en el entrenamiento y predicción no tardo más de 10 segundos

   Desventajas: ´Puede que en este tipo de datos sea funcional pero habria que variar con otros tipos de datos para conocer mejor su rendimiento

   Rendimiento: Excelente con un F1-score promedio de 0.97, igualando a SVM.

Para el conjunto de datos MNIST, los métodos de Random Forest y SVM son preferibles debido a su alto rendimiento en precisión, recall y F1-score. Ambos métodos manejan efectivamente la alta dimensionalidad y la complejidad inherente de los datos de imágenes.
Random Forest es especialmente adecuado si se busca un balance entre rendimiento y resistencia al sobreajuste, siendo además útil cuando se necesita un modelo que pueda manejar múltiples tipos de datos.
SVM es ideal cuando la precisión es crítica y se puede manejar el costo computacional de entrenar con un conjunto de datos grande como MNIST, especialmente cuando se utiliza hardware adecuado o se aplican técnicas para reducir el tiempo de entrenamiento, como la selección inteligente de muestras o la reducción de dimensiones.

PESOS DE VARIABLES

1. Árbol de Decisión (CART)

   Los píxeles más destacados incluyen el 461, 378 y 401, siendo el pixel 378 el más significativo.

   Esto sugiere que estos píxeles, ubicados probablemente en partes estratégicas de la imagen del dígito, juegan un rol crucial en las decisiones de división dentro del árbol.

2. AdaBoost

   AdaBoost resalta los píxeles 434, 571 y 406 como los más influyentes.

   La importancia asignada a estos píxeles por AdaBoost sugiere que la modificación de los pesos de los clasificadores débiles en estas áreas específicas aumenta significativamente la precisión de las predicciones.

3. Bagging

   Los píxeles 489, 350 y 435 son los más destacados en el modelo de Bagging.

   Al igual que en otros modelos de ensamble, Bagging identifica píxeles clave que, si bien pueden variar en posición con respecto a otros modelos, indican zonas de alta variabilidad y relevancia para la clasificación.

4. Random Forest

   Similar a Bagging pero con diferentes píxeles resaltados como el 542 y 433, seguido del 350.

   Random Forest no solo identifica píxeles importantes sino que también reduce la varianza y el sobreajuste al decorrelacionar los árboles individuales, enfocándose en diferentes subconjuntos de características.

Aunque cada modelo de ensamble tiene diferentes píxeles destacados, todos concuerdan en que ciertas áreas de la imagen son más críticas para la clasificación. Esto puede deberse a la posición de características distintivas de los dígitos dentro de estas áreas.

APLICACIÓN INTERACTIVA

Se actualizo la aplicación interactiva que se desarrollo en la tarea 3 de este curso, para poder incluir estos metodos para los cuales se uso el método de la maxima votacion para clasificar el numero dibujado.
Se cargaron cada uno de los modelos pasados con cada una de las configuraciones dadas, para ello se corre la siguiente aplicación.
Se hicieron 3 pruebas las cuales se documentan enseguida

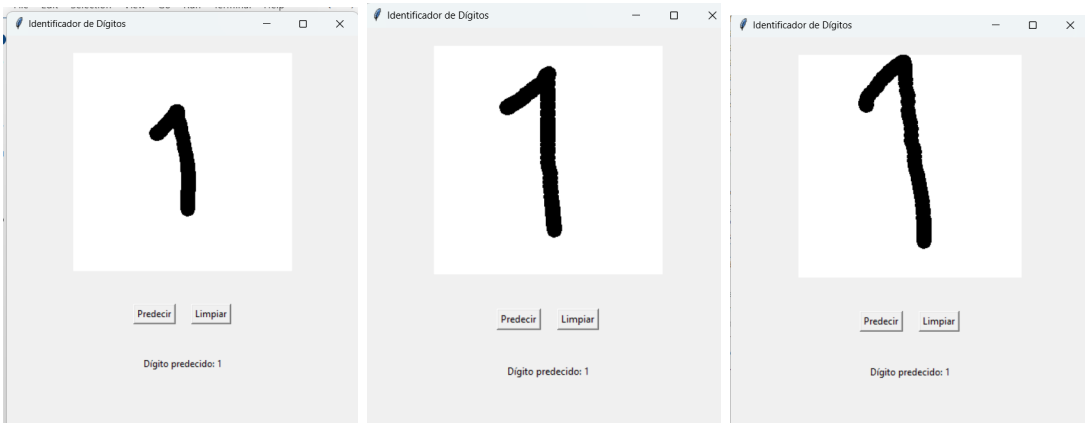

Figura 0.14: Resultados de la applicacion iterativa con el dígito 1

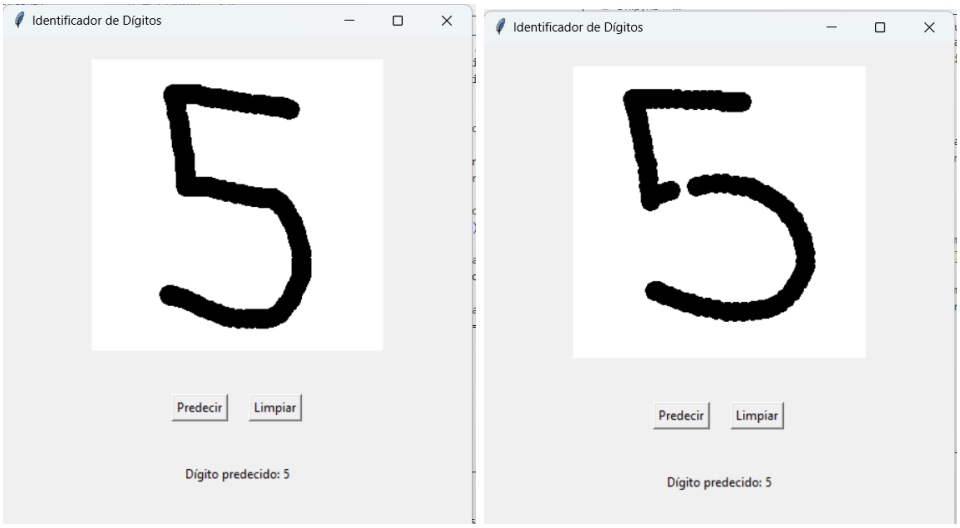

Figura 0.15: Resultados de la applicacion iterativa con el dígito 5

En lo general la aplicación tuvo mucho mejor rendimiento que en la tarea número 3. Muy posiblemente este resultado se deba a que cada modelo en lo individual, tiene muy buen rendimiento todos arriba del 85 %