TP2: Clasificación y validación cruzada

Laboratorio de Datos - 1° cuatrimestre 2023

Grupo: las_yararas
Altamirano Ailen, Rio Francisco, Ruz Veloso Luciano

Introducción

El campo del aprendizaje automático se enfoca en el desarrollo de técnicas y algoritmos que permiten a las computadoras aprender de los datos para tomar decisiones. Dentro de estas, se pueden clasificar en dos enfoques principales, aquellas que son supervisadas y las que no. En el aprendizaje automático supervisado, se incluyen tanto las características de entrada como las de salida, es decir, los datos están etiquetados y se le brinda al modelo cuál es la respuesta esperada. En el no supervisado se espera que el modelo encuentre patrones entre los distintos datos sin proporcionarle información previa sobre lo que se espera de él.

Para el presente informe, se trabajará con el conjunto de imágenes MNIST (Modified National Institute of Standards and Technology), que es muy popular entre quienes se están adentrando al estudio del aprendizaje automático. El dataset de desarrollo cuenta con 60.000 imágenes de 28x28 pixeles, representadas en un archivo CSV donde cada fila es una imagen, la primera columna es el número representado y las siguientes columnas refieren a cada píxel de la imagen.

Dado que este conjunto de datos proporciona etiquetas para cada imagen, se aplicarán técnicas de aprendizaje automático supervisado. En particular, se utilizarán los métodos KNN (k-Nearest Neighbors) y árboles de decisión en conjunto con la validación cruzada para evaluar y validar los modelos.

Explicación de la metodología

La base de datos cuenta con 60.000 filas y 785 columnas. Cada fila representa una imagen, en la que la primera columna señala el número representado y el resto corresponde a cada píxel, donde cada 28 columnas se va avanzando de fila en la imagen.

Lo primero que se puede observar es que los datos se encuentran balanceados (Figura 1), es decir, entre toda la muestra, las cantidades totales de filas correspondientes a cada uno de los números son similares entre sí. Esto es importante ya que puede ayudar a que el modelo tenga mejor rendimiento y evita sesgos que pueden ocurrir cuando alguna clase es dominante por sobre otras (Bishop, 2006, p. 45).

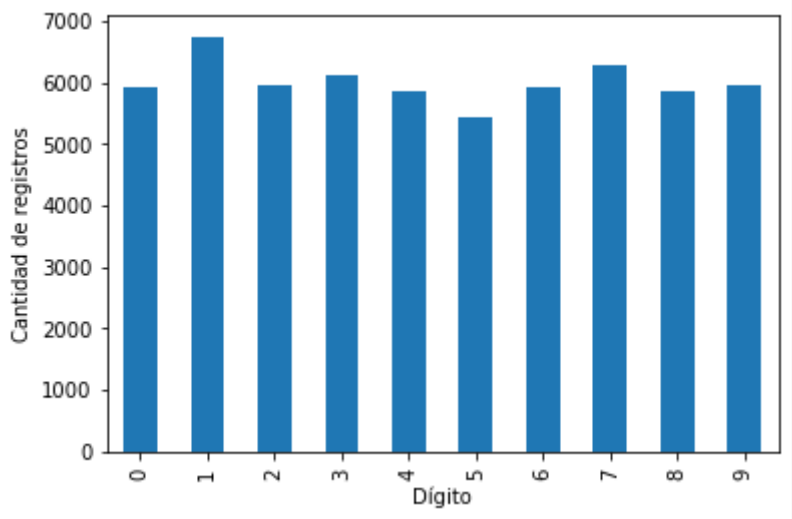

Figura 1. Cantidad de imágenes correspondientes a cada uno de los dígitos.

Luego, la intensidad de cada píxel se representa con un número entre 0 y 250, donde 0 es negro, 250 es blanco y los valores intermedios son escala de grises.

Al mirar imágenes del dataset y hacer los cálculos, se tiene que los bordes (primera y última columna, primera y última fila) no son atributos relevantes a la hora de entrenar algún modelo, debido a que el 99,96% de esos píxeles poseen un valor de intensidad 0. Es decir, independientemente del número representado, la mayoría de esos píxeles presentan intensidad 0, por lo que no aportan información relevante para distinguir entre los distintos dígitos.

Luego de la exploración de los datos, se creó un *subset* que contiene los registros correspondientes a los dígitos cero y uno. Este *subset* contiene 12665 imágenes, correspondiendo el 46,77% a ceros y el 53,23% a unos. Por lo tanto, este subset también se encuentra balanceado, ya que los porcentajes son cercanos al 50%. En estas imágenes se observa que, por lo general, las correspondientes a ceros presentan un centro oscuro mientras que las que corresponden a unos presentan mayor intensidad. En base a esto, exploramos la distribución de la intensidad en distintos píxeles cercanos al centro y en diagonal al mismo según el dígito (cero o uno) al que pertenece la imagen (Figura 2 A y B).

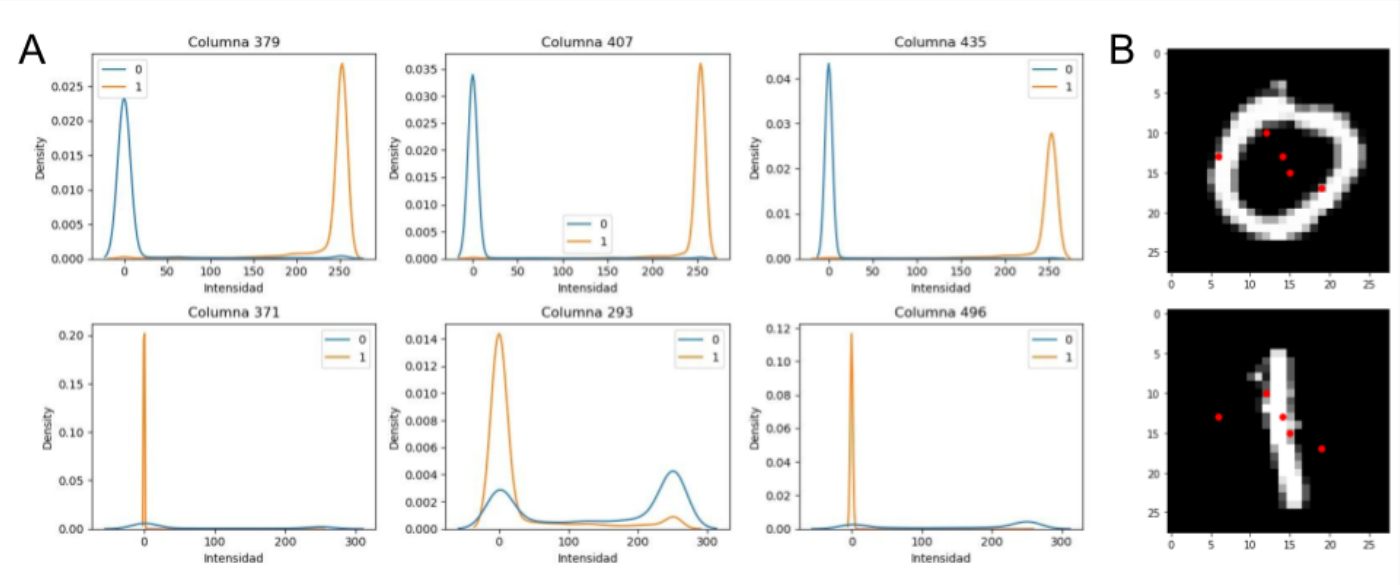

Figura 2. A. Distribución de la intensidad de seis píxeles distintos según el dígito representado. Las distribuciones correspondientes al cero se representan en color azul mientras que las correspondientes al uno se muestran en naranja. El número de píxel corresponde al número de columna indicado en el gráfico menos uno.

B. Representación de dos imágenes, una de cada clase, junto con la ubicación de cinco de los píxeles elegidos en color rojo (columnas 379, 435, 371, 293 y 496).

En las tres figuras superiores (Figura 2A) se observa que los atributos seleccionados permiten diferenciar entre los dos tipos de dígitos, dado que hay poca superposición entre sus distribuciones. En cambio, las distribuciones de intensidad correspondientes a la columna 293 presentan mayor superposición. En cuanto a las columnas 371 y 496, se observa que la clase uno presenta un pico muy agudo en valores de intensidad cercanos a cero.

Una vez realizado esto, se ajustó un modelo de KNN considerando distintas combinaciones de tres, cuatro o cinco de los atributos analizados. Para realizar este análisis se decidió usar un valor de K = 5. Los datos de desarrollo fueron separados en 70% para entrenar al modelo y 30% para validación. Se realizaron cinco repeticiones, y la mayor *accuracy (0.9934)* en *Test* se obtuvo utilizando los atributos correspondientes a las columnas 379, 435, 371, 293 y 496.

Utilizando los atributos con mejor rendimiento, se realizó validación cruzada de cinco particiones para comparar la exactitud de modelos con distintas cantidades de vecinos (K) entre 1 y 14 (Figura 3).

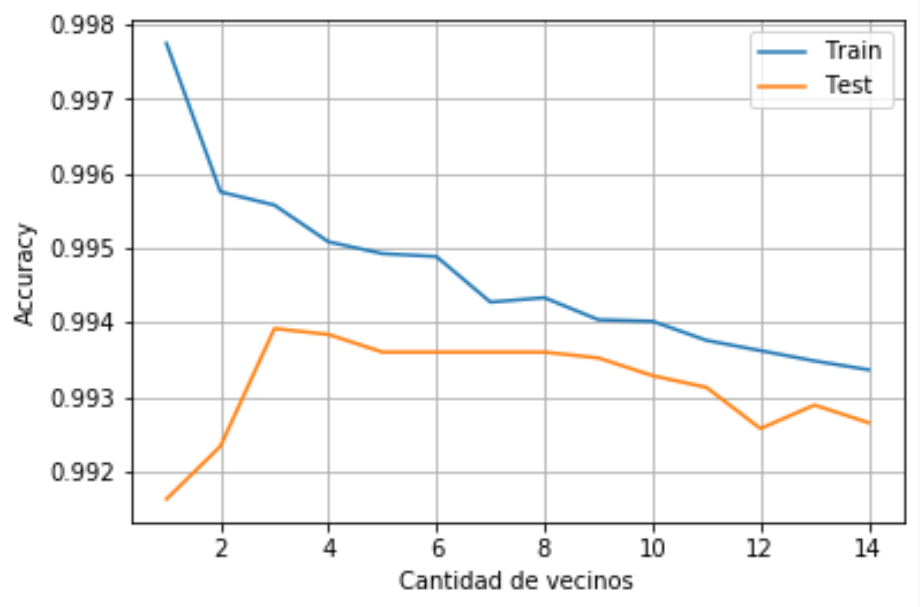

Figura 3. *Accuracy* promedio en función de la cantidad de vecinos. La línea azul representa los datos de entrenamiento (*Train*) y la naranja los de *Test*.

Se observa que el mejor rendimiento de los test se alcanza cuando la cantidad de vecinos es 3 y para cualquier K mayor la *accuracy* promedio va disminuyendo. En cuanto a la *accuracy* calculada usando el conjunto de entrenamiento, se ve que esta disminuye conforme aumenta el valor de K. Elegir un valor de K igual a uno constituye una situación de sobreajuste: la *accuracy* es muy alta para los datos de entrenamiento pero esos resultados no son generalizables, por lo que la *accuracy* en *Test* es menor.

Entrenamos entonces un modelo KNN utilizando K = 3 y los atributos correspondientes a las columnas 379, 435, 371, 293 y 496 utilizando todos los datos de la partición de desarrollo. La *accuracy* obtenida al realizar la evaluación en el conjunto *Held-out* es de: 99.4%.

Luego, para trabajar con árboles de decisión se volvió al dataset original, es decir, con todas las imágenes. En todos los casos, fue posible entrenar al modelo utilizando información de todos los atributos. De no haber sido posible, hubiéramos considerado eliminar los atributos correspondientes a los bordes, debido a lo discutido anteriormente. Se utilizó validación cruzada de cinco particiones para evaluar distintas combinaciones de los parámetros criterio (*entropy* o *gini*) y profundidad (4, 6, 8, 10, 11, 12, 13 y 14). Se graficó el promedio de *accuracy* en función de la profundidad, según el criterio utilizado.

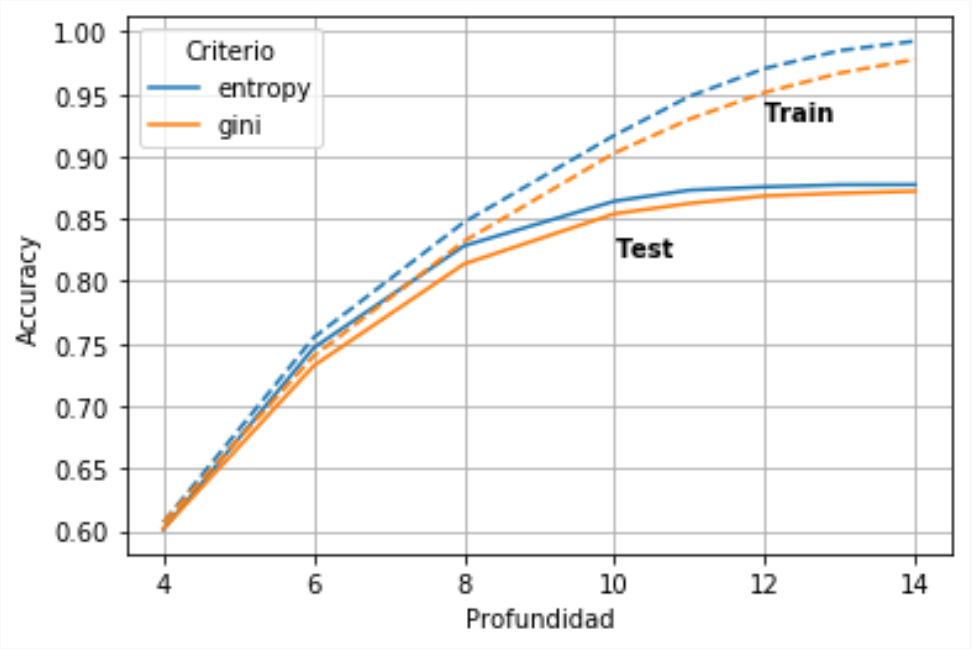

Figura 3. *Accuracy* promedio en función de la profundidad. Los colores azul y naranja sirven para diferenciar entre *entropy* o *gini*, respectivamente. Además las curvas punteadas representan los resultados obtenidos por los datos de entrenamiento, mientras que las otras dos representan al conjunto de test.

Se observa que desde que la profundidad es igual a 4, tanto los datos de entrenamiento como los del test, tienen un rendimiento parecido. Luego, cuando la profundidad es 8, la *accuracy* promedio de Train y Test se van diferenciando de manera más notoria. Luego, en lo que concierne a los datos de test, se tiene que a partir de que la profundidad es 11 el rendimiento deja de aumentar (se alcanza un *plateau*). Para dicho valor de profundidad, la mayor *accuracy* promedio corresponde al criterio *entropy* (entropía) y es mayor a 0,87.

Entrenamos entonces un árbol de decisión con una profundidad de 11 y criterio *entropy* utilizando todos los datos de la partición de desarrollo. La *accuracy* obtenida al realizar la evaluación en el conjunto *Held-out* es de: 88.1%.

Conclusión

Se logró entrenar un modelo KNN con K = 3 utilizando información de cinco píxeles, con una exactitud de 99.4% evaluada en el conjunto *Held-out*. A futuro, podría considerarse entrenar este modelo utilizando mayor cantidad de atributos. Esto no fue realizado en este trabajo dado que con cinco atributos se alcanzó una exactitud en *Test* mayor a 99%, siendo este valor considerablemente alto.

También se logró entrenar un modelo de árbol de decisión con una profundidad de 11 y utilizando la entropía (*entropy*) como criterio, el cual presenta una exactitud de 88.1% evaluada en el conjunto *Held-out*.

Ambos valores de *accuracy* son similares a los promedios obtenidos al realizar la validación cruzada estratificada en el set de desarrollo, lo que demuestra que ambos modelos funcionan de acuerdo a lo esperado a la hora de evaluarlos.

Referencias bibliográficas

- Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.