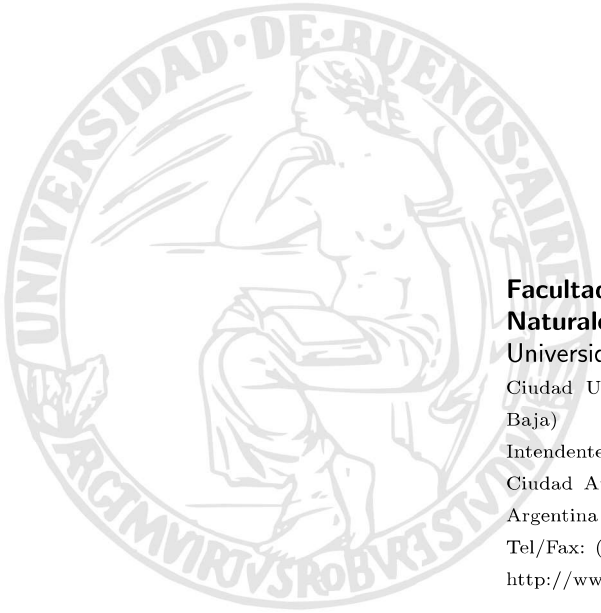
DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 02
Clasificación y validación cruzada

15 Junio de 2022 — Laboratorio de Datos

Integrante	LU	Correo electrónico
Juan Pablo Aquilante	755/18	aquilantejp@outlook.es
Gastón Sanchez	361/22	gasanchez@dc.uba.ar
Mariano Papaleo	848/21	gagopoliscool@gmail.com

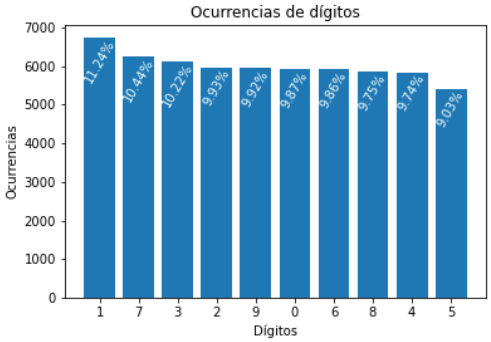


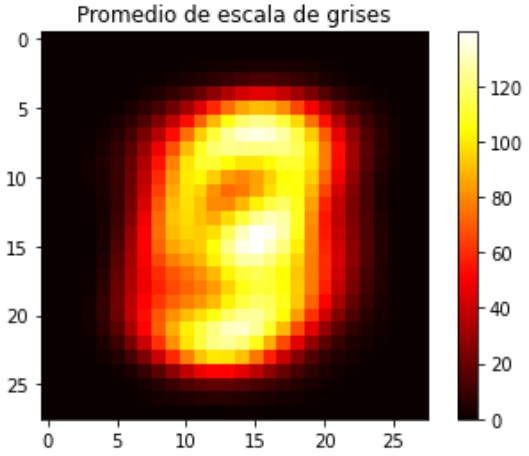
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Introducción

El conjunto de datos consiste en un dataframe de 59999 filas y 785 columnas; cada fila representa una imagen de un dígito del 0 al 9 manuscrito, que fue convertida a escala de grises con una resolución de 28 x 28 píxeles (total 784). En la primera columna de cada fila, aparece la etiqueta de la imagen, y las demás columnas representan cada uno de los 784 píxeles, con un valor del 0 a 255 según el tono de gris correspondiente, 0 siendo blanco y 255 siendo negro. Hay 10 etiquetas (clases) posibles, cada una asociada a cada dígito.

Distribución de datos: En la imagen de la izquierda se puede apreciar las proporciones de los dígitos:





La distribución está balanceada a excepción de los dígitos 1,7 y 5.
Para tener una idea de que atributos/píxeles son los más representativos (los que más se escribieron), se realizó un heatmap en base al promedio de cada columna para las 59999 imágenes. Se puede apreciar en la imagen de la derecha.

Atributos relevantes:
Es evidente que ciertos pixeles van a ser más relevantes que otros a la hora de determinar el dígito en la imagen; los que están en el centro suelen ser escritos más a menudo, y los bordes casi nunca tienen escritura.

Calidad de datos:
También se notó que el DataFrame tenía problemas de calidad de datos. Las columnas tenían nombres poco intuitivos, la columna de etiqueta se llama “5". Decimos cambiarla por “digito" y también las columnas que indican los pixeles fueron renombradas para que representen los 28x28 pixeles.

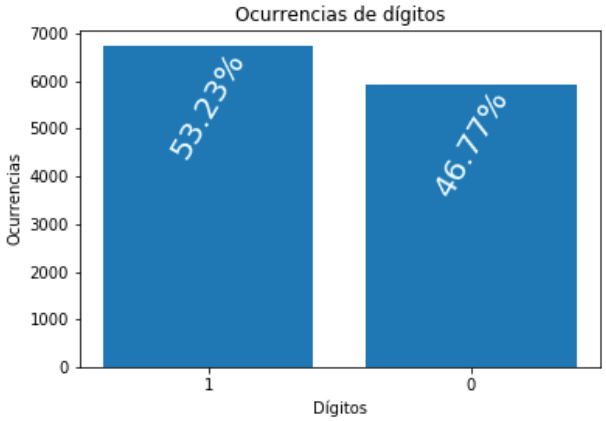
Análisis

Para el dataset de “0" y “1" en nuestro trabajo analizamos 2 casos distintos, el primero es tomar “píxeles representativos" del “0" y del “1", el segundo es tomar “píxeles aleatorios". Se hicieron

análisis primero sobre las imágenes de los dígitos “1" y “0". Filtrándose para obtener los “píxeles representativos".

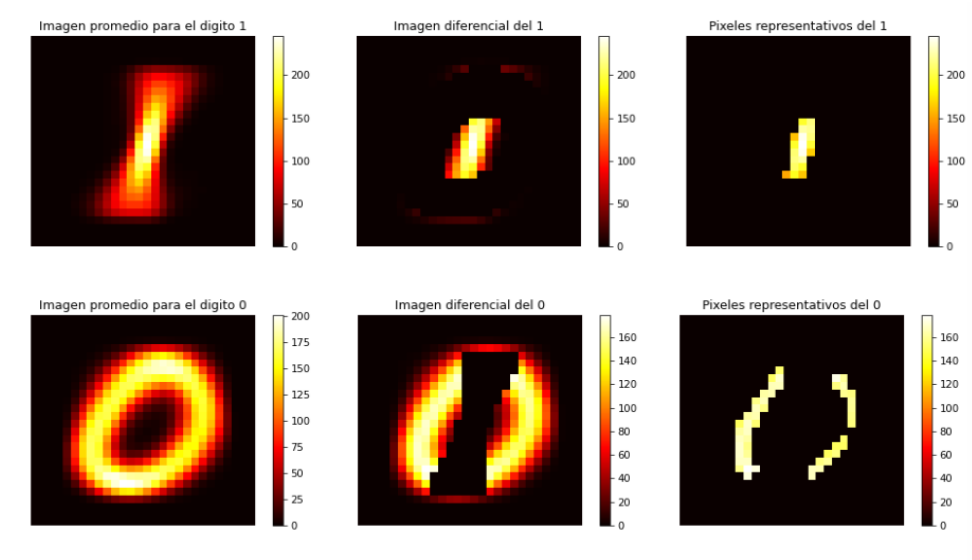
Ejercicio 2 y 3:

Primero se analizaron las ocurrencias de cada dígito; tomando un dataframe con solo las imágenes del “0" y del “1", el “0" forma parte del 46.77% de las ocurrencias, y el “1" el 53.23%, de un total de 12665.



Para determinar qué atributos/píxeles eran los más representativos de un dígito, se hizo el siguiente proceso:

- Se realizaron heatmaps del valor de escala de grises promedio para ambos dígitos.
- Para el “0",se filtró la imagen promedio del “1" para obtener los píxeles más brillantes exclusivos al “0".
- Finalmente, se filtraba la imagen del “0" obtenida otra vez eligiendo los píxeles más brillantes determinado observando el heatmap resultante, obteniendo lo que se denominó la imagen “diferencial" de “0".
- Para el “1", el proceso fue análogo al “0".
- Se obtuvieron 2 arrays con los píxeles más brillantes de cada dígito.

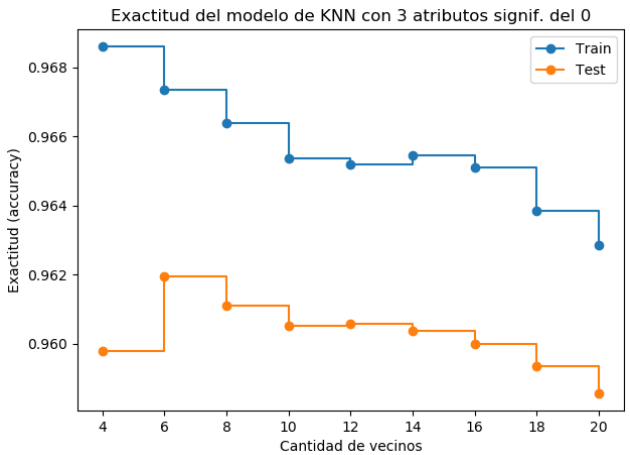


En la cuarta imagen "Imagen promedio (1) diferencial se puede observar los píxeles más brillantes del dígito 1 obtenidos luego de haber sido filtrado al igual que el 0.

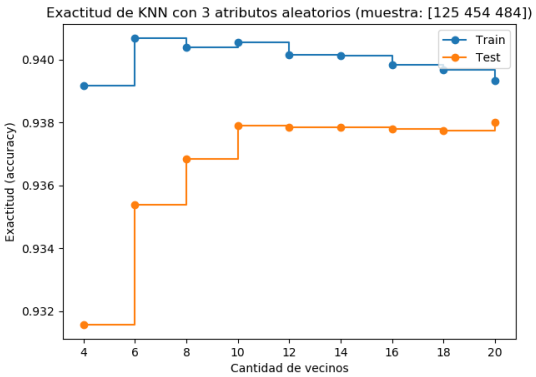
Ejercicio 4 y 5:

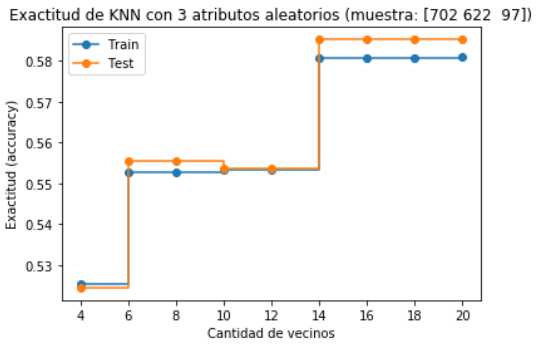
Para el modelo KNN del dataset binario, se hicieron varias tandas de análisis:

- Primero, se eligieron 3 píxeles al azar de los píxeles significativos del 0; la precisión obtenida para los datos de test rondaba consistentemente alrededor del 95%. En la figura de abajo se puede observar un ejemplo de la precisión obtenida.

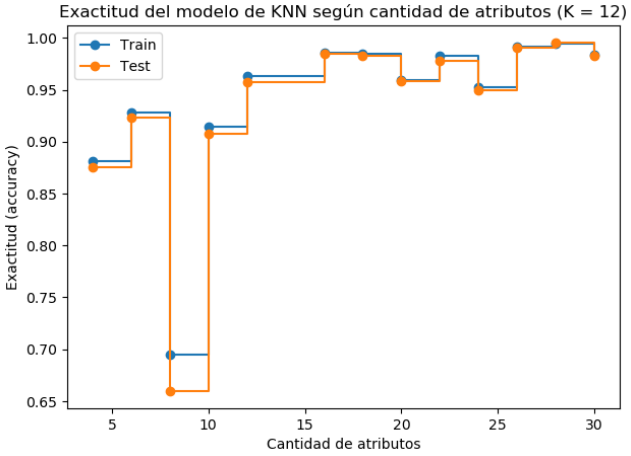


- Luego, se eligieron 3 píxeles al azar de todos los posibles. En la figura inferior se puede observar cómo se obtiene una precisión del 94% cuando uno de los píxeles es significativo (el 484 es significativo del 0). En las demás figuras inferiores, se observan ejemplos para muestras que no tienen píxeles significativos.

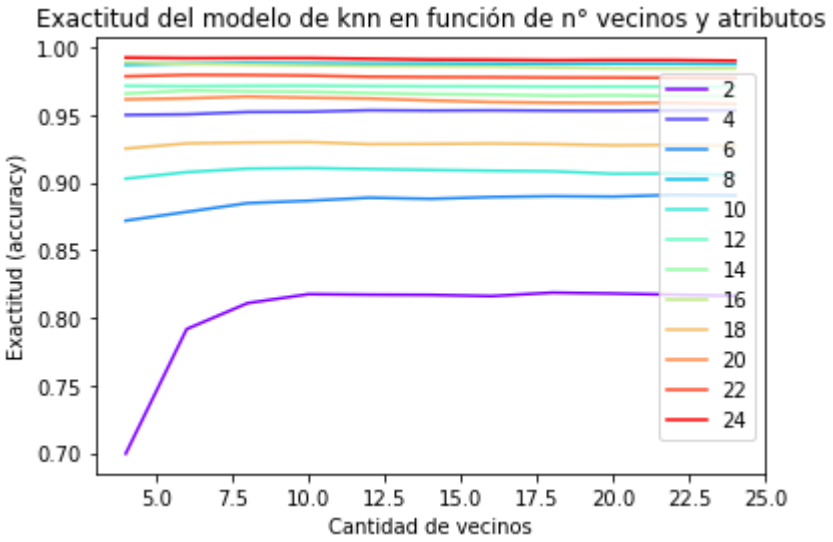




- Después, habiendo fijado el número de vecinos a 12, se probaron usar muestras de varios tamaños. En la figura inferior se muestra la precisión en función del tamaño de la muestra.

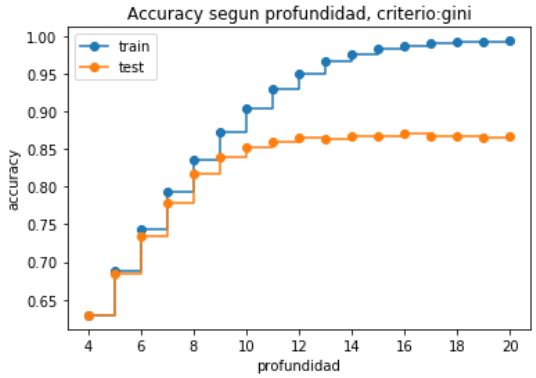


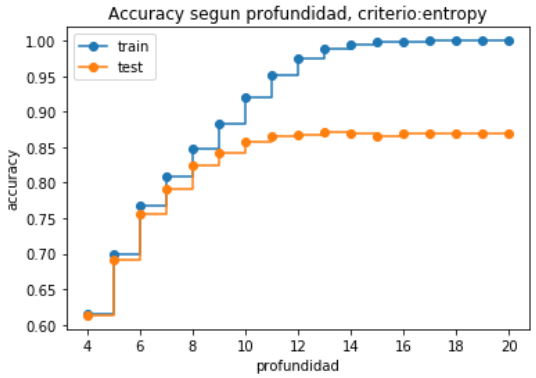
- Una vez que se realizó el trabajo previo de manera exploratoria, se hizo validación cruzada entre los varios modelos obtenidos, variando la cantidad de vecinos y la cantidad de atributos. En la figura inferior se observa como varía la precisión; los vecinos se representan con el eje x y los atributos con los colores.



Ejercicio 6:

Cómo primer enfoque, se utilizaron modelos de árbol de decisión en base a los criterios de entropía y ganancia Gini sin prepruning alguno, dejando la profundidad del árbol libre; se obtuvo una profundidad de 20 con el criterio de entropía y 47 con Gini. En base a esto, se decidió correr los mismos modelos pero ahora fijando la profundidad en 20 para poder comparar ambos modelos. En las figuras de abajo se puede observar la precisión en función de la profundidad para ambos criterios; nótese cómo a partir de cierta cantidad de niveles la ganancia en precisión es despreciable.





Ejercicio 7:

Realizando validación cruzada con k-folding de 5 folds para varios niveles de profundidad del árbol de decisión con criterio de entropía, se determinó que el nivel de profundidad óptimo era alrededor de 14-16. Más allá de esto, también se observó que los puntajes de validación cruzada obtenidos a partir de 8 niveles de profundidad oscilaban alrededor de 97.5%.

Usando como profundidad 14 en el árbol de decisión y también con validación cruzada con k-folding, también se hizo una comparación con el método de KNN para 12 vecinos, usando todos los atributos posibles.

Conclusiones

KNN de 3 o más atributos, dataset binario (ej. 4, ej. 5)

En el ejercicio 4, se observó que no era sólo importante la cantidad de atributos sino cuáles eran los que se elegían; la precisión obtenida en muestras de 3 atributos al azar era altamente variable, dependiendo de si los atributos encontrados eran píxeles significativos del dígito 0 o 1. Teniendo en cuenta esto y eligiendo los píxeles significativos de cualquiera de los dos dígitos, se conseguía precisión alrededor del 95% de forma más consistente. También se observó que, habiendo fijado el número de vecinos, incrementando el tamaño de las muestras aleatorias daba mejor performance, dado que se tiene mayor probabilidad de acercarse a los píxeles significativos.

También, se determinó observando la precisión en función de los atributos elegidos al azar (con vecinos fijos a 3) que el número óptimo de atributos era alrededor de 12-15; a partir de ese punto, el incremento en precisión era despreciable y en ciertos casos hasta empeoraba.

En el ejercicio 5, para hacer un análisis de cómo varía la precisión con ambos parámetros (los vecinos y los atributos) en conjunto, se hizo validación cruzada entre los distintos modelos resultantes. En promedio se observaba lo esperado: en promedio, mientras más vecinos y atributos se tenían, mejor era la precisión. Cómo cantidad óptima de vecinos, también se encontró que rondaba alrededor de 10-15; a partir de ese punto, las ganancias en precisión eran despreciables.

Árboles de decisión sobre todos los dígitos, validación cruzada con K-folding (ej.6, ej. 7)

Del ejercicio 6 se pudo observar que no limitar la profundidad de un árbol de decisión resultó en overfitting; habiendo pasado 12-14 niveles de profundidad el train score se aleja mucho del test score, y este último no aumenta. En base a esto se decidió como profundidad óptima del árbol de decisión el 14. También se comparó el desempeño entre los árboles con criterios de ganancia Gini y entropía; entropía le ganó a Gini por una diferencia ínfima, entonces se decidió para el siguiente ejercicio tomar como criterio entropía.

En el ejercicio 7, al contrastar el árbol de decisión con una profundidad de 14 contra un KNN con 12 vecinos, y ajustar ambos modelos con K-Folding Cross Validation usando todos los atributos del dataframe, se observó que el KNN posee una mayor accuracy en promedio que el árbol de decisión a la hora de predecir datos. Sin embargo, el árbol sigue teniendo una accuracy bastante elevada pero no mayor al KNN.

Testeo de modelos sobre el holdout provisto (ej.8)

Para el ejercicio 8, los tests que se nos fueron asignados para evaluar la accuracy de los modelos que entrenamos a lo largo de este trabajo. Obtuvimos lo siguiente:

1. En el DataFrame para testear con todos los dígitos evaluamos tanto con el modelo del Árbol de decisión como el KNN.
   - Con Árbol de decisión obtuvimos una precisión que oscilaba entre 86-88% en cada iteración con una profundidad de 14 y con todos los atributos del DataFrame de entrenamiento (ya que el Árbol es muy rápido para entrenarse).

   - Con KNN obtuvimos una precisión que oscilaba entre 96-97% en cada iteración con 12 vecinos y 12 atributos elegidos aleatoriamente del DataFrame de entrenamiento.

   - Estos datos se corresponden con nuestros resultados y conclusiones que obtuvimos del ejercicio 7.

2. En el DataFrame binario para testear el dígito 0 y 1, evaluamos con el modelo KNN

- Con KNN obtuvimos una precisión muy alta de 97-99% al diferenciar los dígitos binarios utilizando **píxeles significativos del cero**.
- Al diferenciar los dígitos binarios utilizando **pixeles significativos del uno,** también obtuvimos una precisión de 97-99%.
- Como bien concluimos en el ejercicio 5, al utilizar **píxeles significativos**, aunque la cantidad de vecinos y/o atributos utilizados para entrenar el modelo sea baja, el accuracy es alto de igual forma. Para probar esto al testear el df_binario entrenamos al modelo KNN con vecinos k = 5 y atributos = 3. Relativamente bajos y la precisión fue la antes mencionada 97-99%.
- Por último testeamos el df_test_binario con **píxeles significativos no aleatorios**, pero en este caso si son necesarios una cantidad óptima de vecinos y atributos. Por ellos utilizamos un estándar de modelo KNN con vecinos k = 15 y atributos = 15. Los resultados fueron mejor de lo esperados. El Test Score oscila entre 90% y 99%, como toma 15 atributos aleatorios de los 784 píxeles, a veces toma píxeles muy poco representativos de los dígitos, o que pueden llegar a confundir dígitos. Y eso hace que las predicciones puedan llegar a ser malas pero eso se compensa con el hecho de que lo entrenamos con una alta cantidad de vecinos, en particular, k = 15. Desde luego que tomando 15 de forma aleatoria de un DataFrame que en total tiene 784 elementos y predecir con un accuracy mayor a 90% e incluso que puede llegar a ser 99%, es decir, con un promedio mayor a 90% nos parecen resultados que exceden nuestras hipótesis.