Trabajo Práctico: Clasificación y validación cruzada

Rostan Marcos; Dramis Agustín

Laboratorio de datos - 1er. Cuatrimestre 2024

Departamento de computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires

Introducción

Se trabajó con el conjunto de datos de EMNIST, que consiste en imágenes de letras en imprenta mayúscula escritas a mano. Estas imágenes están representadas como filas en data frame, donde la primera columna “Letras” es la variable de interés y las demás columnas, los atributos, representan la intensidad del trazo en cada píxel de una imagen de 28 x 28 px.

El objetivo fue implementar modelos de aprendizaje automático, haciendo una evaluación y selección con criterio de dichos modelos.

Análisis exploratorio

La base contiene registros de 62.400 imágenes correspondientes a las 26 del alfabeto latino. Los registros se encuentran uniformemente distribuidos entre las distintas letras (Figura 1). Al visualizar el promedio de todas las imágenes disponibles, se pudo observar la existencia de una “zona útil” en el centro de la imagen, y bordes poco utilizados en el trazo (Figura 2). Esto sugiere la posibilidad de que ciertos atributos de la tabla, es decir ciertos pixeles de la imagen, aportan poca información a la hora de diferenciar las letras entre sí. Aquellos atributos que son consistentemente 0 podrían ser descartados a la hora de desarrollar métodos de clasificación, ya que un atributo debe varias para integrar un criterio de clasificación.

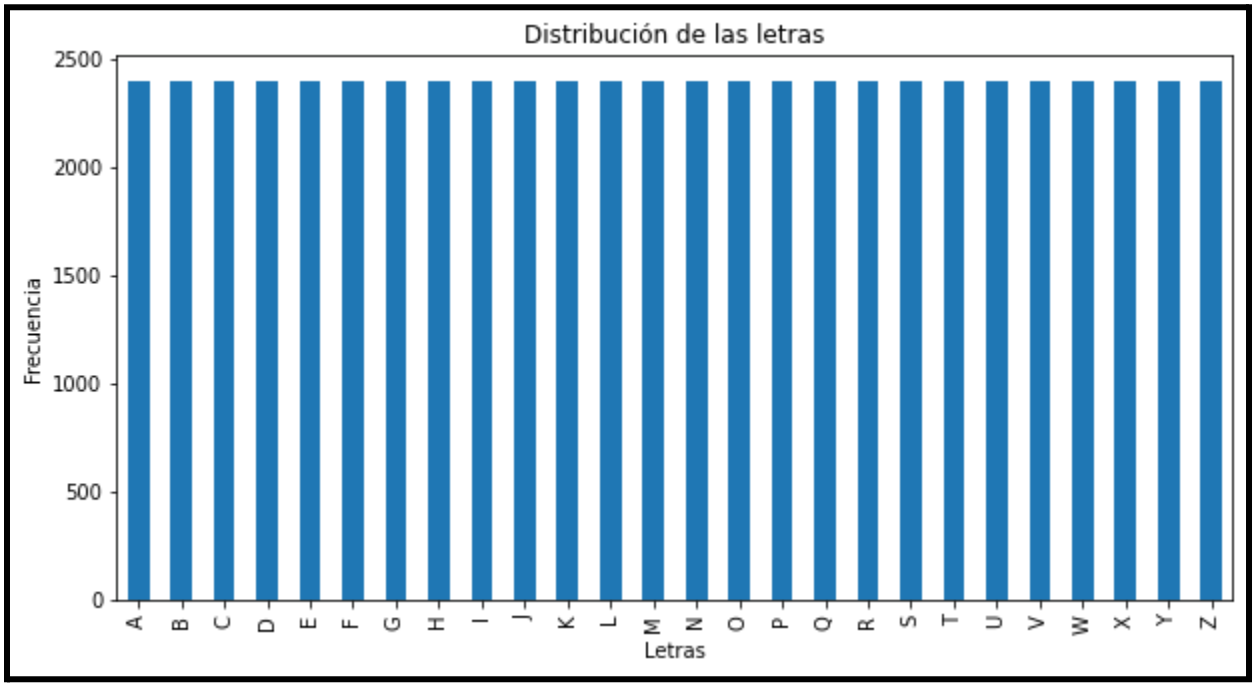

Figura 1. Frecuencia de aparición de cada letra en la base de datos.

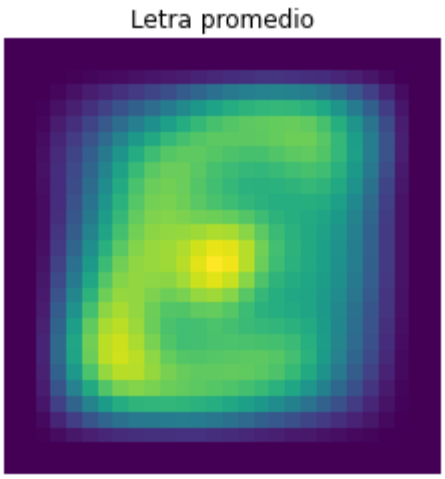

Figura 2. Imagen del promedio de todas las imágenes

Para analizar la “consistencia” de las letras, es decir el nivel de similitud entre distintas imágenes correspondientes a la misma letra, Se visualizó la dinámica del desvío estándar, utilizando la visualización de letras. Para ello, se visualizó una fila compuesta por los desvíos estándar correspondientes a cada pixel (Figura 3). Al analizar la letra “C”, a modo de ejemplo, se observó una zona de baja variabilidad con en el centro del trazo de la letra, mostrando cierta consistencia en las imágenes, rodeada de un borde de mayor variabilidad (Figura 4). Esta forma de “letra hueca” muestra que el centro del trazo guarda cierta consistencia intra-letra.

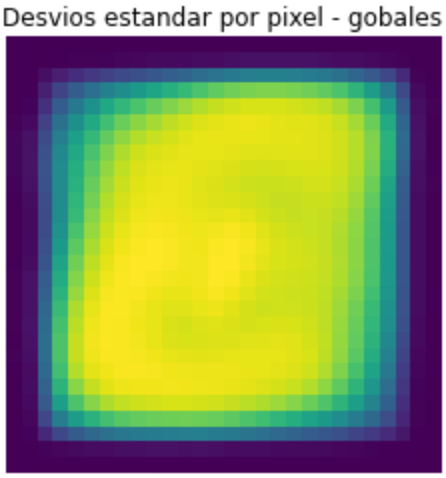

Figura 3. Representación visual de los desvíos estándar de cada pixel.

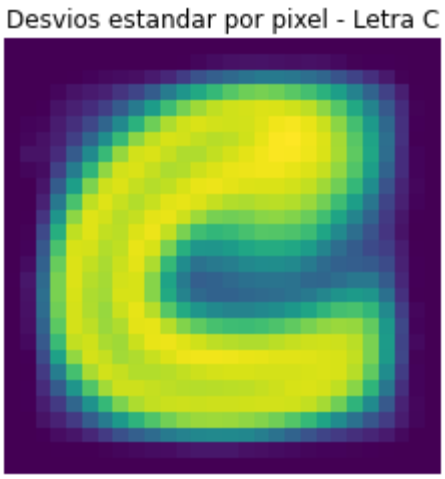

Figura 4. Representación visual de los desvíos estándar de cada pixel para imágenes de la letra “C”.

Para analizar la similitud entre distintas letras, se tomaron los atributos no universalmente nulos, es decir aquellos en los que al menos una fila presentó un valor distinto de 0. Sobre estos valores, se calcularon las letras “E”, “L” y “M” promedio. Luego se graficaron las intensidades de cada pixel en una letra en función de las intensidades en la otra. El par E - M mostró un patrón desordenado, evidenciando la baja similitud entre las letras (Figura 5). Por otro lado, el par E - L (Figura 6) mostró un patrón más definido, con una mayor cercanía a la diagonal, evidenciando una mayor similitud entre las letras, debido a la presencia de líneas coincidentes en su figura.

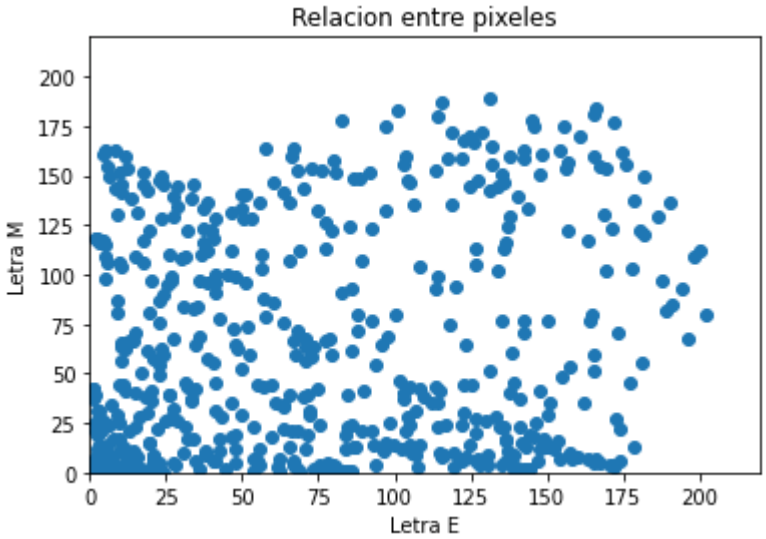

Figura 5 - Intensidades de los pixeles no nulos en la letra M y en la letra E.

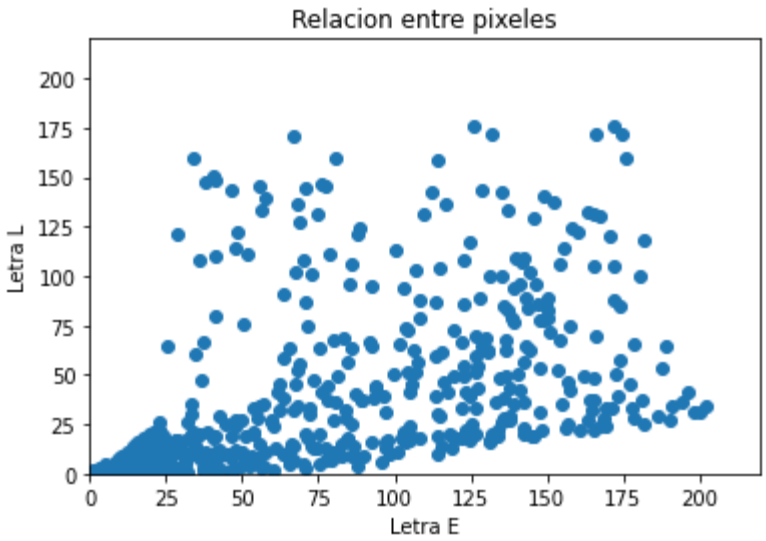

Figura 6 - Intensidades de los pixeles no nulos en la letra L y en la letra E.

Clasificación binaria

Se tuvo el desafío de poder predecir si una letra era una A o una L mediante la intensidad de sus pixeles. Para eso se buscó un modelo KNN tal que su exactitud a la hora de predecir esté cercana al 100%.

Para encontrar ese modelo, se separaron los datos totales del subconjunto que contenía a las muestras de las letras A y L en un 80% train y 20% test para después poder evaluar cada modelo.

Luego, se ajustaron modelos KNN de 3 vecinos (k=3) con 5 conjuntos de diferentes cantidades de atributos (3, 10, 50, 100, 300, 500) y se analizó la exactitud por cantidad de atributos (figura 7).

En donde se pudo apreciar que con 50 atributos se llegaba a un buen modelo en términos de exactitud, ya que era indistinguible la diferencia con los de mayor cantidad de atributos teniendo en cuenta el costo computacional.

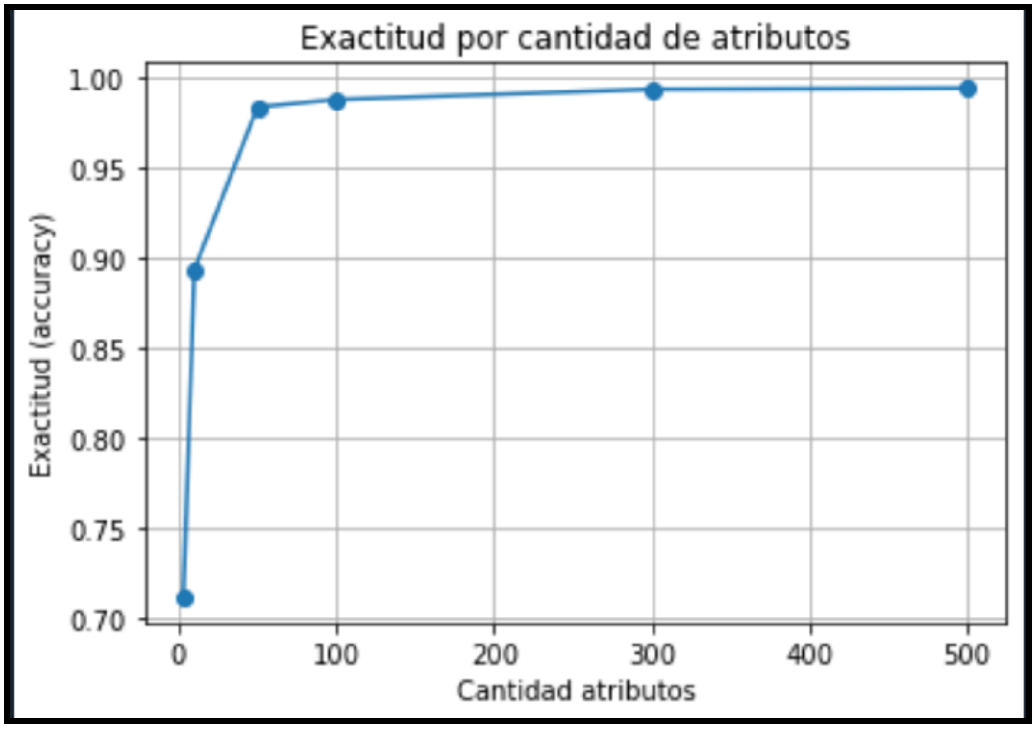

Figura 7. Exactitud en función de cantidad de atributos con 3 vecinos.

Posteriormente, para lograr un mejor modelo, se compararon modelos variando no solamente la cantidad de atributos sino que también el número de vecinos “k”. Utilizando los mismos 5 conjuntos de diferentes cantidades de atributos que anteriormente pero variando el “k” de 1 a 50 vecinos. Se evaluaron las exactitudes promedio por cantidad de atributos y cantidad de vecinos (figura 8).

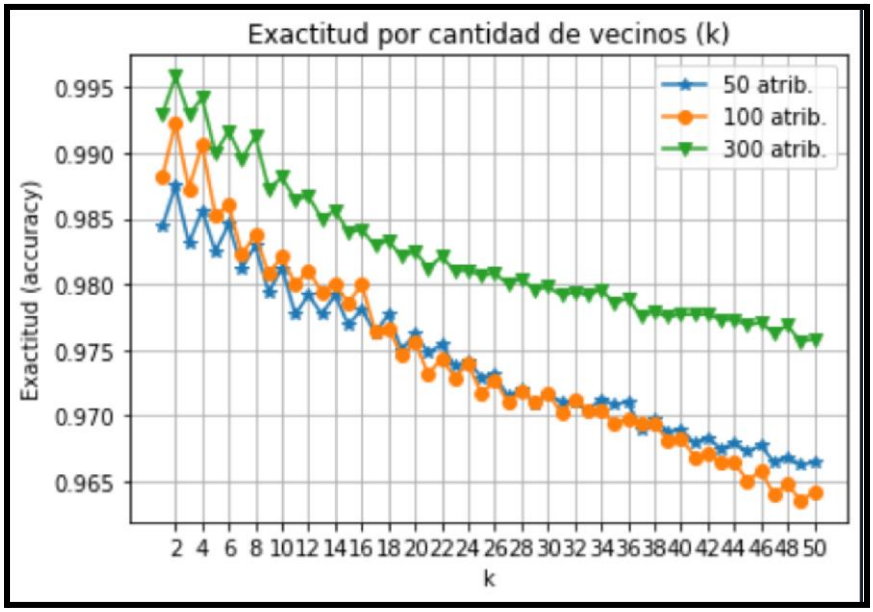

Figura 8. Exactitud por cantidad de vecinos para los conjuntos de 50,100 y 300 atributos

En donde se puede apreciar que la cantidad de vecinos k=2 es donde se obtiene el modelo con mejor rendimiento.

Clasificación multiclase

Dada una imagen, el objetivo fue predecir qué vocal era. Para ello, primero se separó el conjunto de datos que contenían solamente a las vocales en un 80% desarrollo y el otro 20% en validación.

Luego, se ajustaron modelos de árbol de decisión probando con diferentes profundidades, a nuestro criterio usamos arbitralmente estas (1,2,3,5,10,13,15, 20, 25, 45), y para poder tomar la decisión de qué profundidad era la mejor, se gráfico la exactitud del modelo en función del hiperpárametro (figura 9).

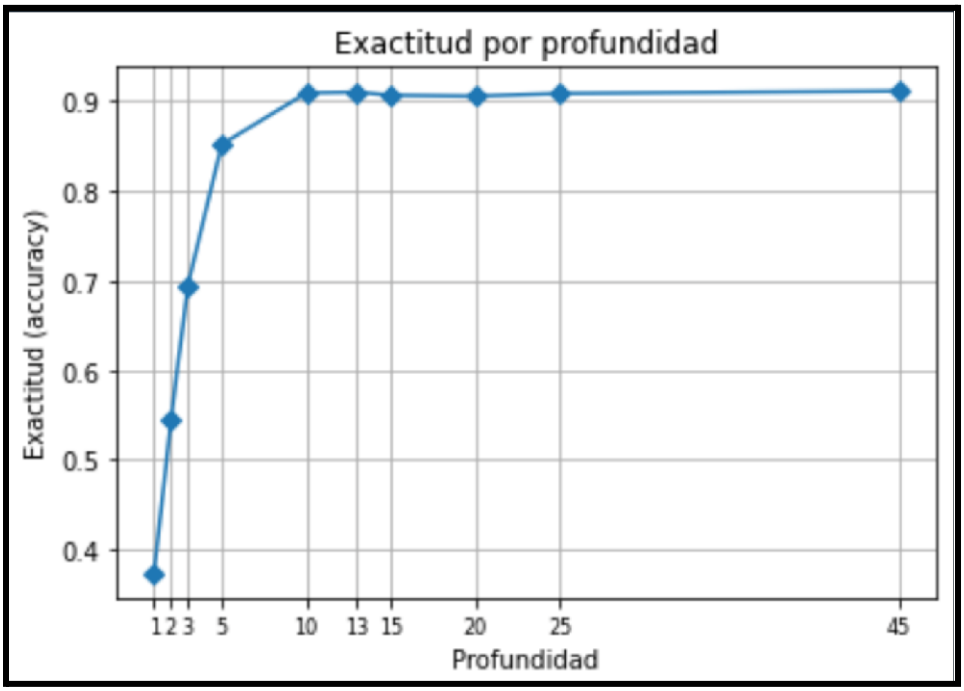

Figura 9. Exactitudes de los modelos de árbol de decisión en función de su profundidad.

Cómo se logra apreciar en el gráfico, la profundidad que mejor ajusta a los datos es 10.

Posteriormente, se realizó un experimento en el cual se buscaba el modelo que mejor ajustaba a los datos haciendo distintas combinaciones de hiperparametros (nosotros elegimos max_depth, max_feature y criterion). Utilizando validación cruzada con K-folding se obtuvo el mejor modelo.

Luego se entrenó dicho modelo en todo el conjunto de datos de desarrollo y se utilizó para predecir las clases en el conjunto held-out. Su performance está dada por la matriz confusión de la figura 10.

429	9	4	9	15
18	446	4	9	11
5	8	466	1	5
11	9	2	471	22
14	9	4	14	405

Figura 10. Matriz confusión del modelo luego de predecir en el conjunto held-out.

La información que se puede recolectar de dicha matriz confusión se ve resumida en la sección conclusiones.

Conclusiones

Este trabajo práctico se basó en 3 grandes pilares, la exploración de los datos, la clasificación binaria y la clasificación multiclase. En la parte de la exploración de los datos se presentó una gran dificultad a la hora de poder tomar conclusiones y diferencias entre las letras, ya que no es algo intuitivo recolectar información sobre pixeles.

En cambio, en la parte de ambas clasificaciones se logró obtener modelos con buenos resultados, debido a la posibilidad de abstraerse de la intensidad de los pixeles y solo mirarlos como atributos.

Resultados de la clasificación multiclase: figura 11 y figura 12

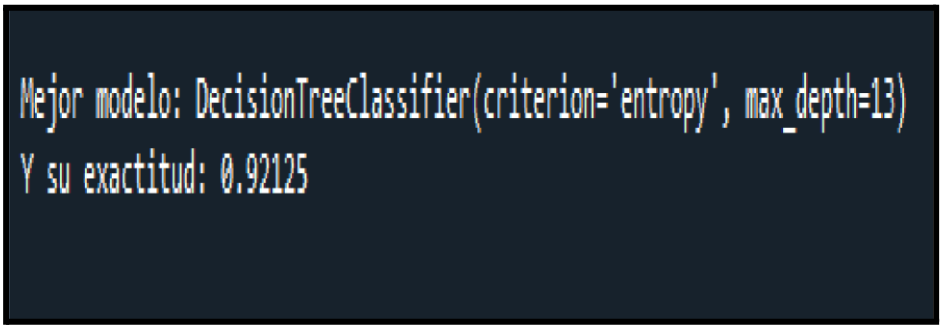

Figura 11. El mejor modelo del experimento en clasificación multiclase y su exactitud

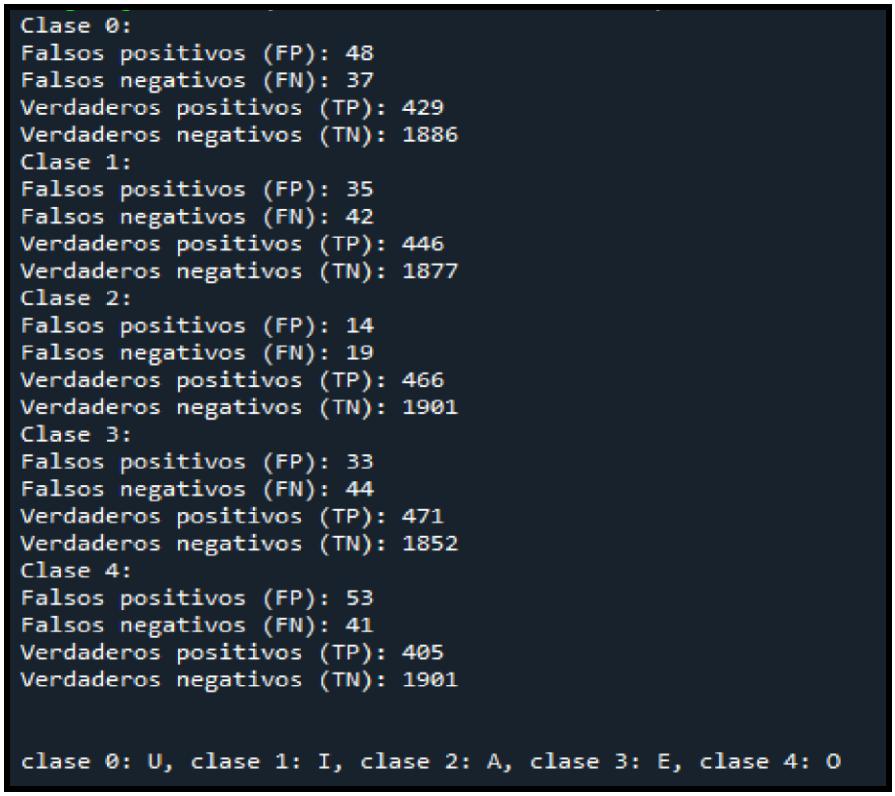

Figura 12. cada clase y sus distintos tipos de errores