Laboratorio de datos
TP-02

Grupo MLJ
Otermín Juana, Quispe Rojas Luis Enrique, Vilcovsky Maia
Facultad de ciencias exactas y naturales

A. Introducción y análisis explorativo

Se trabajó con un DataFrame compuesto por imágenes. El mismo contaba con 62.400 filas x 785 columnas. Cada fila representa una letra y cada columna un píxel de esa letra. Se observó que el mismo posee todas las letras del abecedario exceptuando la letra "Ñ", que cada letra se repite 2400 veces pero cada imagen es distinta entre sí, es decir, el DataFrame nos muestra 2400 formas de escribir un tipo de letra.

Este DataFrame planteó un nuevo desafío, ya que trabajar con imágenes requiere un algoritmo como mediador para visualizar y analizar los datos. Es distinto a los DataFrames que veníamos trabajando, en los cuales podíamos explorar los datos de forma directa y sacar conclusiones sin necesidad de intermediarios, simplemente observando los datos o utilizando herramientas como pandas o SQL.

Con el objetivo de encontrar los atributos relevantes para predecir una letra se realizó un análisis en donde se promedio cada columna del DataFrame y se realizó una nueva imagen, la cual vendría a ser el promedio de todas las letras. Ver figura 1.

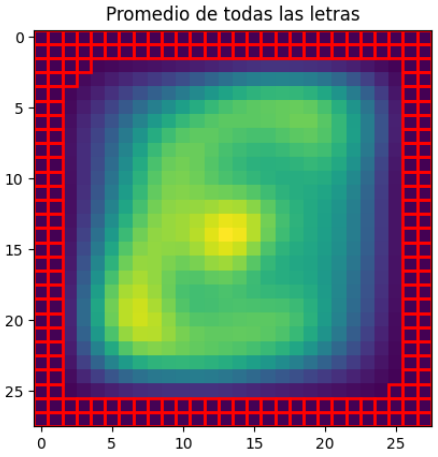

Figura 1: Promedio de todas las letras. Los cuadrados en color rojo representan los píxeles que se pueden descartar.

Se tomó como criterio que si el promedio del píxel es menor a 1 no es relevante para distinguir la letra, y por lo tanto se puede descartar. Esto se representa con un cuadradito rojo en la imagen.

Si se observa la figura 1 se puede notar como el promedio de todas las letras parece formar la letra "E", esto se puede deber a que la letra E posee una estructura simétrica, posee líneas horizontales y verticales, esto es un factor predominante en muchas letras, por ejemplo las letras "F", "L", "T", "I", "R", "P", "D", "H", "K", "B" y "N" interceptan la letra "E".

Con el objetivo de averiguar si hay letras parecidas entre sí se realizaron dos análisis. El primero fue promediar cada letra en una nueva imagen, es decir, se tomaron las 2.400 imágenes que representaban una misma letra y se promedió en una nueva imagen (Figura 2).

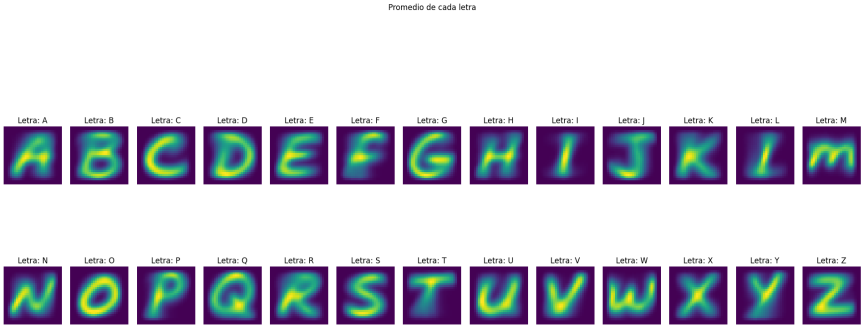

Figura 2: Promedio de cada letra.

Luego de observar esta imagen se concluyó que los posibles pares de letras parecidos son: "L" - "I", "M" - "N", "E" - "F", "O" - "Q", " X" - "V", "B" - "D", "C" - "U", "E" - "M", "E" - "L". Para verificar esto se realizó el segundo análisis (Figura 3). Se calculó la distancia euclídea para dichas letras (Ver Ec 1 donde A y B son las matrices que representan la imagen promedio de una letra), se normalizó tomando un criterio y se realizó un heatmap.

$$d = \sqrt{\sum_{i=1}^{784} (A_i - B_i)^2} \tag{1}$$

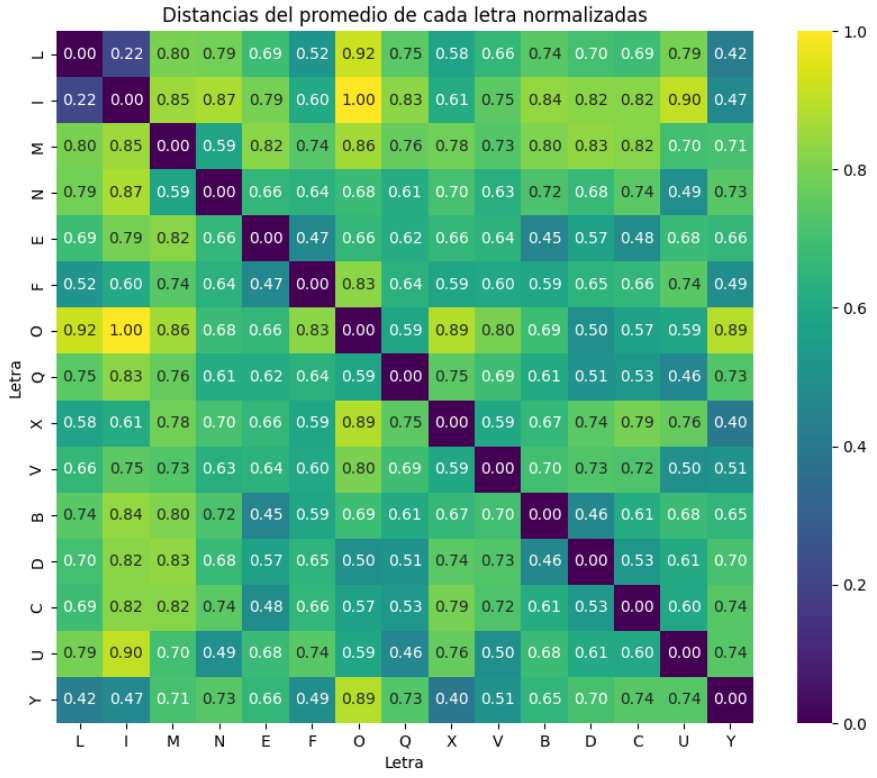

Figura 3: Distancia entre las letras seleccionadas, cada una es el promedio de su clase. Los colores oscuros representan menor distancia (mas similitud), los colores claros representan mayor distancia (menos similitud).

Considerando que los colores oscuros representan distancias más cortas y los colores claros representan distancias más largas, se pudo concluir que la “L” y la “I” son las letras más similares, con una distancia de 0,22. En contraste, la “O” y la “I” son las letras más diferentes entre sí, con una distancia de 1. Al observar la figura, podemos notar que la “E” es más similar a la “L” que a la “M”, con distancias de 0,69 y 0,82, respectivamente.

Nos resultó interesante preguntarnos cuál es la letra que más se parece a las demás. Para esto, realizamos el mismo análisis de la figura 3, pero para todas las letras, y promediamos cada fila. Descubrimos que las letras que más se asemejan a las demás son la “E”, la “F” y la “K”. Esto se podría explicar la figura 1, en la que se observa la “E” como predominante, ya que la letra “F” parece fusionarse con la “E”, restando importancia a la letra “K”.

Para determinar la similitud entre las imágenes de la clase “C”, se calculó el desvio standard promedio de la columna para cada letra. Luego, se promediaron las desviaciones estándar para obtener un valor representativo de cada una. Esto nos sirvió para poder ver cuán dispersas están las imágenes de la letra “C” en relación con su imagen promedio. Los resultados se presentaron en un gráfico de barras (Ver figura 4), donde la desviación estándar de la letra “C” se destacó con una línea punteada roja, facilitando así la comparación.

Se concluyó que la desviación estándar promedio de la letra “C” es pequeña en comparación con las demás letras, lo que indica que las imágenes de la letra “C” son bastante

similares entre sí. Sin embargo, las imágenes de la letra “I” resultaron ser aún más similares, presentando la menor desviación estándar promedio. En contraste, se observó que la letra “B” presenta la mayor variabilidad en sus imágenes, ya que su desviación estándar promedio fue la más alta.

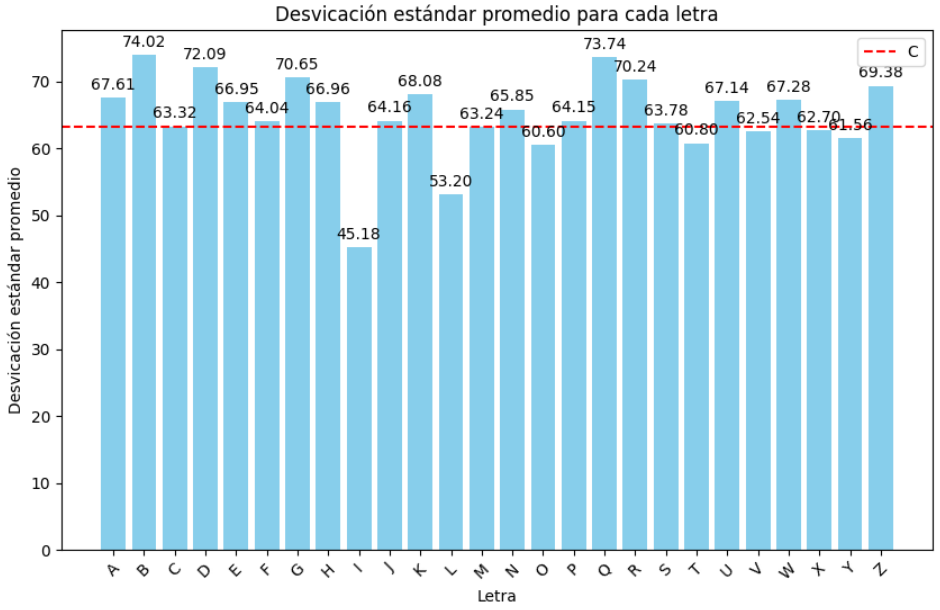

Figura 4: Gráfico de barras de la desviación estándar promedio para cada una de las letras.

B. Experimentos y análisis realizados

B.1. Clasificación binaria

Para poder hacer una clasificación binaria entre la letra “A” y la letra “L”, se creó un nuevo DataFrame con todas las clases de estas dos letras (mezcladas) y para que el análisis sea correcto y esté balanceado, se verificó que haya la misma cantidad de clases de cada una (2400 muestras por letra).

El DataFrame se dividió en 80 % para entrenamiento y 20 % para validación. El experimento comenzó seleccionando 3 atributos basados en las mayores distancias promedio entre las letras “A” y “L” (dentro de un rango de distancia entre 150 a 188.67 (máxima distancia)). Esta selección maximiza la varianza de los datos, mejorando así los resultados.

Para visualizar esto, se creó un boxplot mostrando la distribución de los valores para las clases “A” y “L” en la columna seleccionada (Ver figura 5). Como se observa, los valores de los píxeles para las clases “A” son mucho más altos que para las clases “L”, cuya media es cercana a 0, con solo algunos valores atípicos visibles. Un ejemplo de los pixeles que tomamos se pueden visualizar mejor en la figura 6.

Este comportamiento da mucha información para el modelo ”*KNN*”, ya que una columna con una diferencia tan grande entre las clases ayuda a distinguir entre “A” y “L” de manera efectiva. Por eso se decidió tomar este criterio para continuar con el modelo de clasificación.

En el experimento, se seleccionaron 3 atributos dentro de los que presentaban mayores distancias, observando un accuracy elevado que mejoraba aún más cuando tomabamos atributos que no eran consecutivos, y esto se debe a que cada píxel no es independiente por sí solo, ya que no se produce un cambio de valor abrupto sino que va variando gradualmente.

Posteriormente, se aumentó la cantidad de atributos (5, 10 y 15 atributos de las máximas distancias) y se notó que el accuracy mejoraba a medida que estos aumentaban. Esto se debe a que cada atributo adicional puede contener información relevante, ayudando al modelo a distinguir mejor entre las letras y a determinar la similitud entre los puntos con mayor precisión. Con pocos atributos, el modelo puede no capturar todas las variaciones y patrones presentes en los datos.

Finalmente, se tomó el modelo de 15 atributos y se variaron los valores de k, es decir, la cantidad de vecinos. La métrica utilizada fue el accuracy, cuyos resultados se muestran en la figura 7. Se observó que con k=1 la precisión del conjunto de entrenamiento es muy alta, pero esto es normal ya que cada punto es su propio vecino más cercano, llevando a un sobreajuste (overfitting) y lo que se refleja en un alto accuracy también para el test. A medida que k aumenta, la precisión del conjunto de test aumenta rápidamente y alcanza un máximo alrededor de k=5 lo que sugiere que este valor de k proporciona un buen equilibrio para el modelo. Después de este punto, la precisión del conjunto de test muestra algunos cambios pero tiende a disminuir lentamente a medida que k sigue aumentando, esto se debe a que el modelo se vuelve demasiado general y empieza a subajustar.

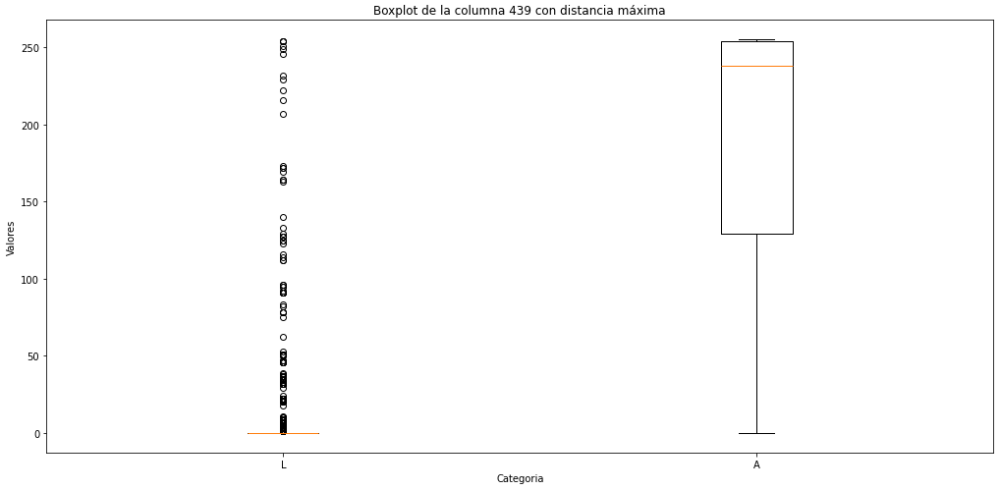

Figura 5: Boxplot de la distribución de valores para las clases “A” y “L” en la columna con máxima distancia entre ellas.

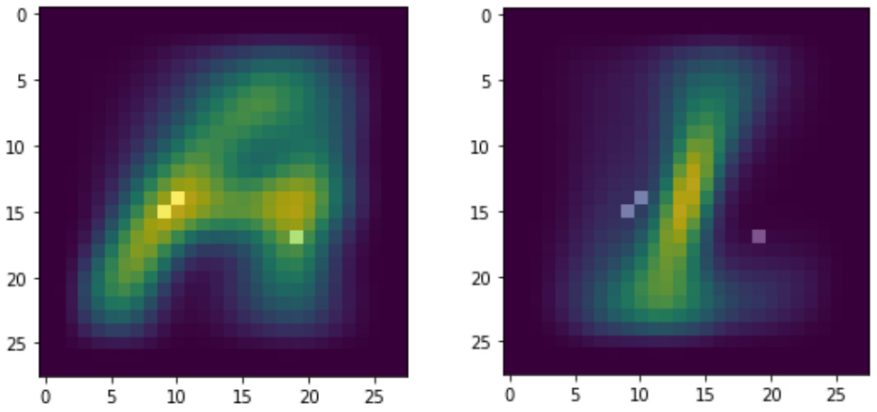

Figura 6: Promedios de la letra “A” y “L” seleccionando los 3 pixeles donde se dan las máximas distancias entre ellas.

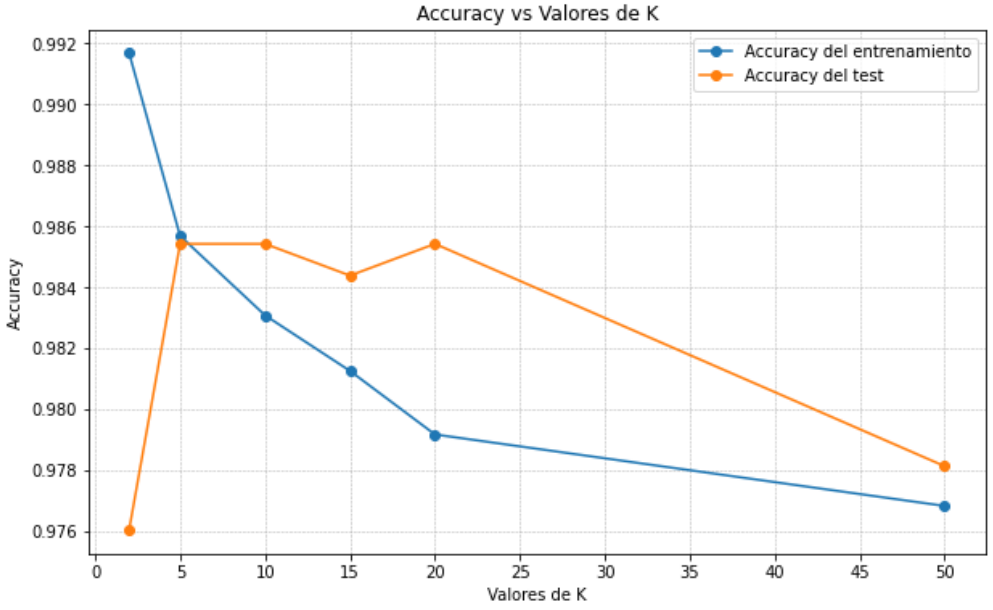

Figura 7: Gráfico de variación del accuracy vs los valores de k para el modelo de K-Nearest Neighbors (KNN).

B.2. Clasificación Multiclase

Se buscó clasificar las imágenes correspondientes a las vocales mediante el uso de un algoritmo de árbol de decisión. En primera instancia se filtraron los datos correspondientes a las vocales verificando que los mismos estén balanceados, luego el DataFrame fue dividido en 80 % para desarrollo y 20 % para validación (held out).

Se probaron distintas profundidades y distintas medidas de impureza haciendo cross validation (en el conjunto train) y usando como medida de evaluación el accuracy para ver

cual de estas combinaciones era la óptima. Esto se muestra en la figura 8, en donde se ve que el accuracy va subiendo a medida que la profundidad del árbol va aumentando, llegando a estancarse para valores por encima de 7 aproximadamente, para ambas medidas de impureza. Como criterio para encontrar la profundidad óptima que no complejiza innecesariamente el modelo, se tomó como profundidad 4, ya que en ese punto la variación respecto a sus próximos vecinos no es tan considerable. Además se eligió como medida de impureza la entropía ya que su accuracy era mayor al de gini.

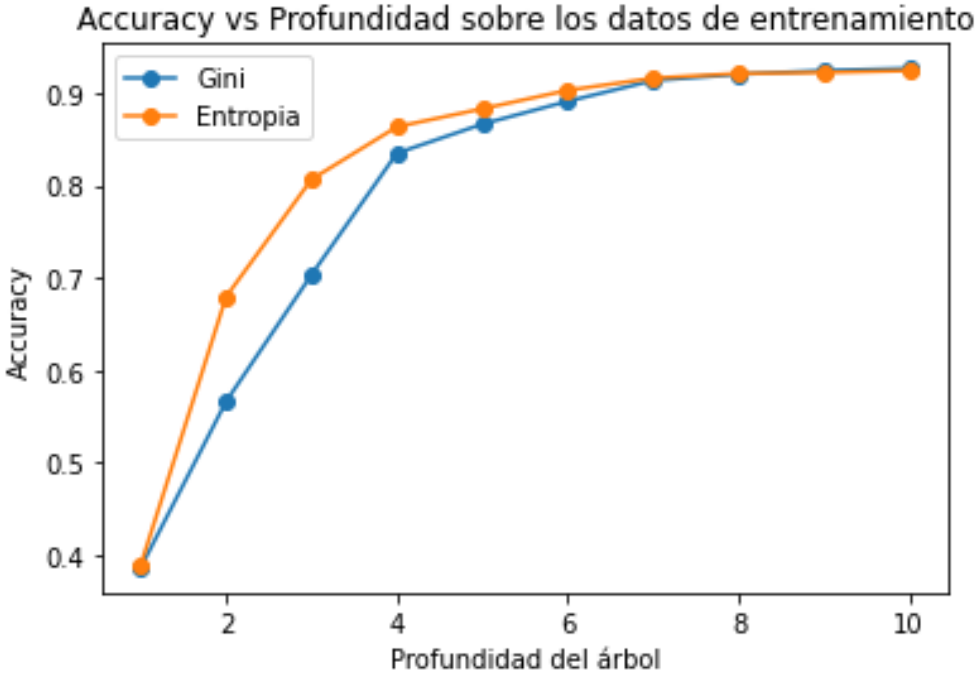

Figura 8: Gráfico de accuracy vs profundidad para el modelo de árbol de decisión sobre los datos de entrenamiento (train).

La figura 9 muestra el esquema del modelo elegido (reentrenado con los datos de train) con altura 4 y medida de impureza entropia. Este modelo se utilizó para predecir las clases del conjunto held-out. En donde para evaluar el desempeño del modelo se utilizaron distintas métricas a partir de la matriz de confusión graficada en la figura 10 y que se resumen en la tabla 1.

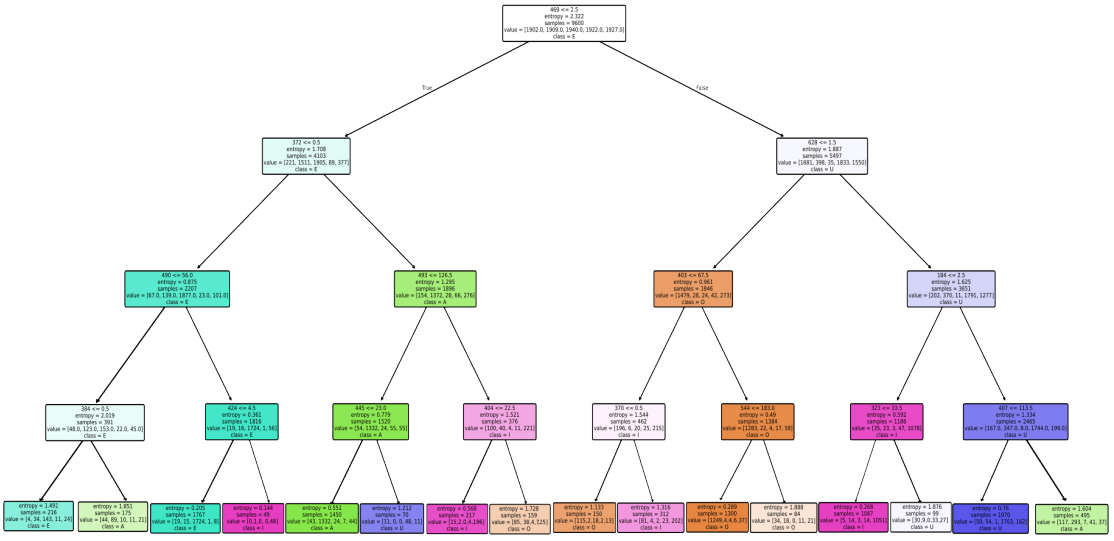

Figura 9: Esquema de árbol de decisión

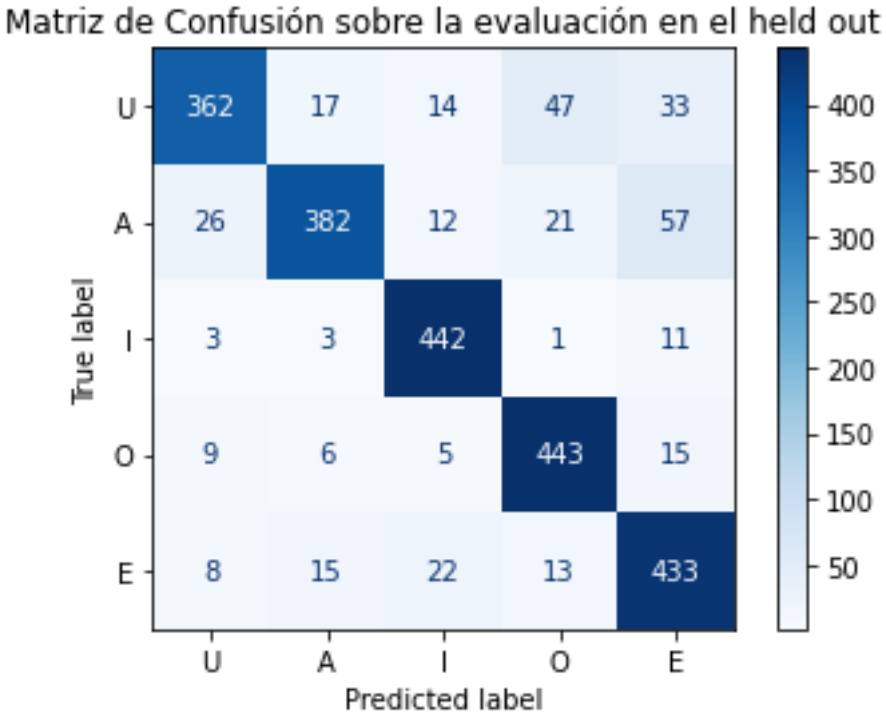

Figura 10: Matriz de Confusión para la vocales según el resultado de evaluar los datos en el held out.

Métrica	Valor
Accuracy	0.8588
Precision	0.8627
Recall	0.8599
F1 Score	0.8582

Cuadro 1: Resultados de las Métricas de Evaluación

De la figura 10 se observa que el modelo no presenta dificultad para diferenciar entre “O” e “I”. En la figura 11 se muestra los distintos pixeles que el árbol analiza para determinar si se trata de una estas dos vocales, para un camino particular. Por ejemplo para esta rama en concreto realiza [469,628,403,544] para la letra “O” y [469,628,403,370] para la letra “I” donde los números referidos son los píxeles analizados. Se puede observar que son los mismos salvo el último que permite diferenciar entre ambos conjuntos.

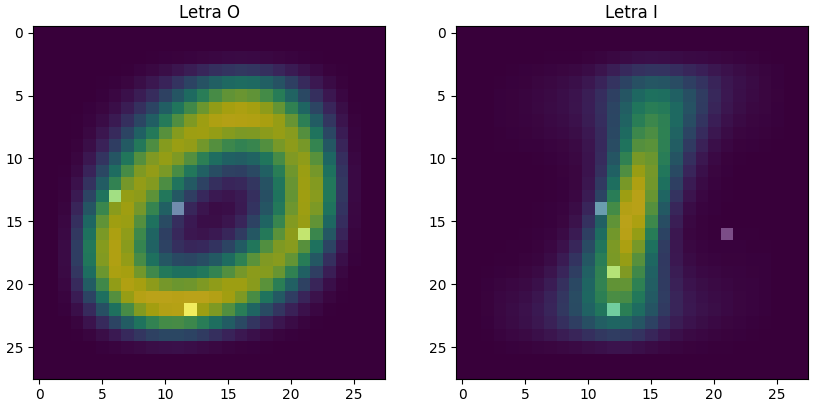

Figura 11: Comparación entre pixeles analizados para letra “O” e “I”.

En general las métricas de la tabla 1 son muy cercanas entre sí, lo que sugiere un modelo bien equilibrado. La similitud entre precisión y recall indica que el modelo no solo está prediciendo bien las instancias positivas, sino que también está evitando una gran cantidad de falsos negativos. Un F1 Score similar a la precisión y el recall refuerza la idea de que hay un buen balance y no hay un sesgo hacia falsos negativos.

C. Conclusión

Se concluyó que las letras que más se asemejan a las demás son la “E”, la “F” y la “K”. Y al hacer el promedio de todas las letras aparece la letra “E” como predominante. Se concluyó, utilizando como criterio la distancia euclídea, que “L” e “I” son las más similares entre si, mientras que “O” e “I” son las más diferentes.

En la clasificación binaria entre las letras “A” y “L” usando K-Nearest Neighbors (KNN), se seleccionaron atributos basados en las mayores distancias promedio, mejorando la precisión del modelo al incrementar la cantidad de atributos y ajustar el valor de “k” a cinco.

Para la clasificación multiclase de las vocales usando como modelo un árbol de decisión, se determinó que la profundidad óptima del árbol es cinco, y la entropía es la medida de impureza más adecuada con una validación cruzada de 10 particiones.

El modelo mostró un buen desempeño en el conjunto de validación. La matriz de confusión reveló que el modelo distingue bien entre las vocales, especialmente entre "O" e "I", con mínimos falsos positivos. Las métricas obtenidas a partir de la matriz de confusión son muy cercanas entre sí, lo que sugiere un modelo bien equilibrado. La similitud entre precisión y recall indica que el modelo no solo predice correctamente las instancias positivas, sino que también minimiza los falsos negativos.