DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico N°2

Clasificación y validación cruzada

4 de Junio de 2024 — Laboratorio De Datos

Grupo 4

Integrante	LU	Correo electrónico
Suarez Ines	890/22	ine.suarez22@gmail.com
Ramirez Ana	931/23	correodeanar@gmail.com
Wittmund Montero, Lourdes	1103/22	lourdesmonterochiara@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

1. Introducción

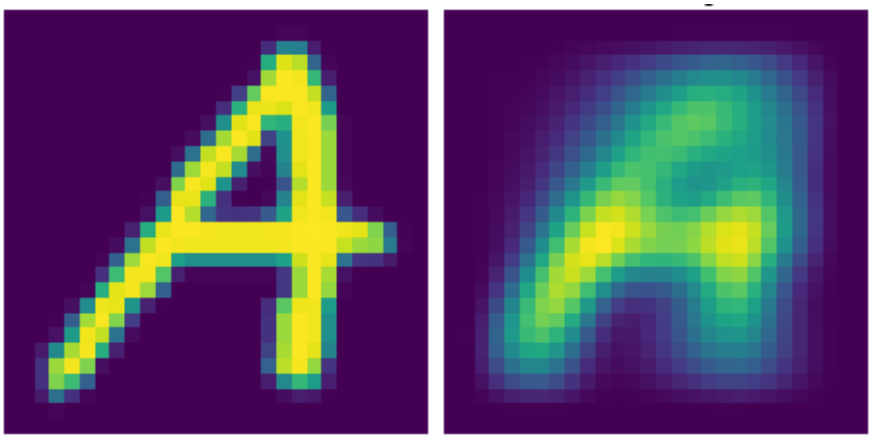
En el ámbito del aprendizaje automático, la clasificación de datos es un proceso fundamental que permite asignar categorías a diferentes instancias en función de ciertas características. En este sentido, la validación cruzada es una herramienta crucial para evaluar y comparar el comportamiento de diversos modelos predictivos. El objetivo principal de este trabajo práctico es evaluar los conocimientos adquiridos en cuanto a clasificación y selección de modelos. Para ello, nos fue brindado del conjunto de datos EMNIST un subconjunto que corresponde a las letras mayúsculas. A lo largo de este informe, exploraremos diversas estrategias de análisis y modelado para entender mejor la naturaleza de los datos y determinar qué enfoques son más efectivos para la clasificación de las letras representadas en las imágenes.

El conjunto de datos EMNIST es un conjunto de dígitos de caracteres escritos a mano derivados de la Base de Datos Especial NIST 19 y convertidos a un formato de imagen de 28x28 píxeles y una estructura de conjunto de datos que coincide directamente con el conjunto de datos MNIST. Cada pixel de cada imagen esta representado con un número del 0 al 255, siendo el 0 el color negro y el 255 el blanco.

2. Análisis exploratorio

Para realizar un análisis exploratorio de estos datos es necesario evaluar diversos aspectos, como la cantidad total de datos disponibles, el número y tipo de atributos presentes, la cantidad de clases de la variable de interés (letras), y otras características relevantes que se consideren importantes. Por eso queremos responder las siguientes preguntas:

1. **¿Cuáles parecen ser atributos relevantes para predecir la letra a la que corresponde la imagen? ¿Cuáles no? ¿Creen que se pueden descartar atributos?**

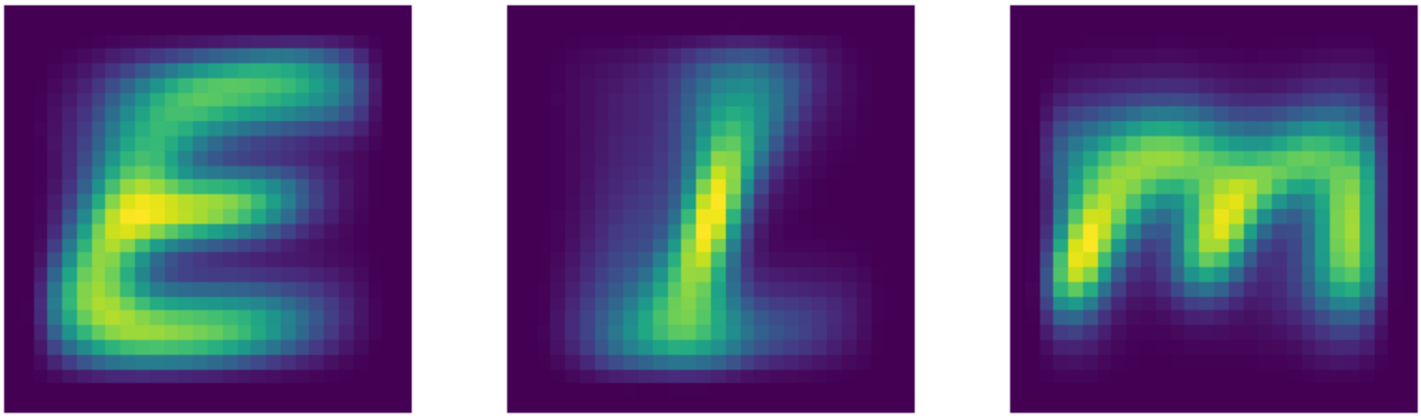


(a) Imagen letra A (b) Imagen promedio letra A

Figura 1

Para ilustrar el concepto de los atributos relevantes, hemos optado por seleccionar una letra al azar, en este caso la A, y generar una imagen promedio utilizando todas las imágenes disponibles para esta letra. En 1a, se muestra una imagen aleatoria de la letra A, mientras que en 1b se presenta la imagen promedio. En esta última, los píxeles que permanecen en tonos violetas incluso después de incorporar todas las posibles variaciones de las imágenes de la letra A, son considerados como atributos irrelevantes, ya que no aportan información sobre cómo está escrita la letra. De igual manera, los píxeles muy cercanos al violeta podrían considerarse irrelevantes por la misma razón. Estos píxeles mencionados podrían ser descartados sin afectar la capacidad de distinguir la letra representada en la imagen. Por el contrario, los píxeles en tonos amarillos y celestes destacados parecen ser altamente relevantes para predecir la letra, ya que son consistentes en la mayoría de las imágenes de la misma letra. Por lo tanto, estos píxeles son fundamentales para definir la identidad de esa letra sin lugar a dudas.

2. **¿Hay letras que son parecidas entre sí? Por ejemplo, ¿Qué es más fácil de diferenciar: las imágenes correspondientes a la letra E de las correspondientes a la L, o la letra E de la M?**



(a) Imagen promedio letra E (b) Imagen promedio letra L (c) Imagen promedio letra M

Figura 2

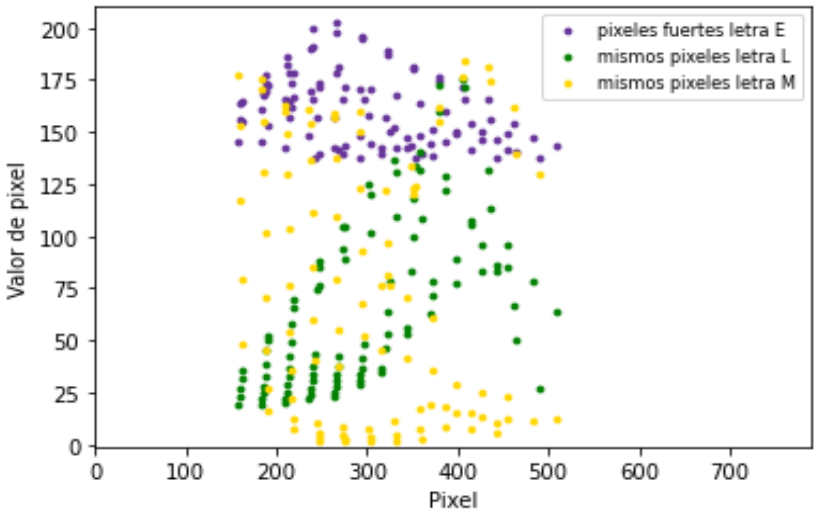
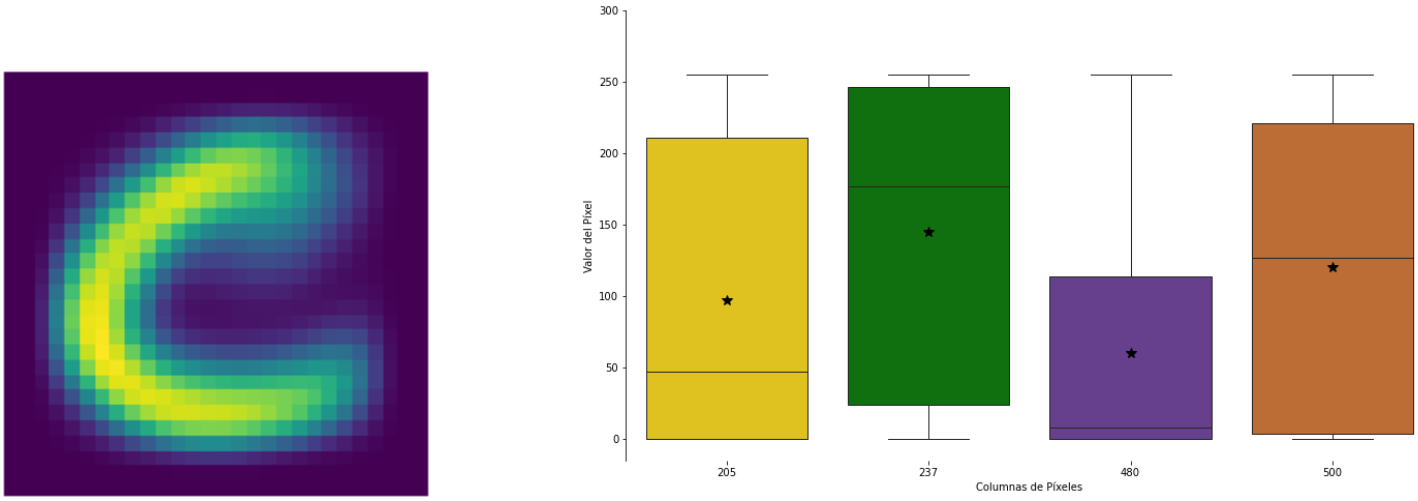


Figura 3: Diferencias de las letras E , L y M en los píxeles mas fuertes de E

En la Figura 3, podemos observar la comparación de las imágenes promedio de las letras E, M y L en términos de los píxeles más fuertes de la letra E. Cada punto violeta en el gráfico representa un píxel fuerte de E y los puntos amarillos y verdes representan esos mismos píxeles pero de las letras M y L respectivamente. Además, la coordenada X indica el píxel y la coordenada Y indica el valor que toma este pixel en la escala de color de 0 a 255. Se observa que los puntos violeta y amarillo se superponen en varios casos, en cambio, los puntos verdes están más separados. Esto sugiere que los píxeles más fuertes en promedio de la letra E tienen valores más cercanos a los de la letra M que a los de la letra L, lo que puede ser útil en tareas de reconocimiento de patrones o clasificación de letras ya que ambas letras comparten tonalidades parecidas al ser graficadas gracias a tener parecidos valores de pixeles. Basándonos en esto, podemos concluir que la letra E parece más fácil de diferenciar con la letra L que con la letra M.

3. **Tomen una de las clases, por ejemplo la letra C, ¿Son todas las imágenes muy similares entre sí?**

Primero graficamos la imagen promedio de la letra C para distinguir similitudes y píxeles más o menos fuertes. Luego, para poder responder la pregunta, analizamos el valor de todas las diferentes imágenes correspondientes a la letra C en cuatro píxeles específicos. Para ello, realizamos un gráfico de boxplot.



(a) Imagen promedio letra C

(b) Variación de 4 píxeles específicos en imágenes de la letra C

Figura 4

Como se puede observar, todos los box, excepto el del píxel 480, tienen una gran amplitud, lo que refleja variabilidad entre las imágenes. El píxel 480 es el que menos varía, como tiene la mediana tan cerca del 0, la mayoría de las imágenes tienen ese valor en ese píxel; sin embargo, para todos los píxeles elegidos hay al menos una imagen con el valor mínimo posible (0) y otra con el valor máximo posible (255). Este gráfico parece indicar que las imágenes pueden no ser tan similares entre sí; todo depende de a qué píxeles se preste atención.

4. **Este dataset está compuesto por imágenes, esto plantea una diferencia frente a los datos que utilizamos en las clases (por ejemplo, el dataset de Titanic). ¿Creen que esto complica la exploración de los datos?**

   Sí, creemos que trabajar con un dataset compuesto por imágenes complica la exploración de los datos en comparación con los datasets que usamos en las clases como el del Titanic. Esta complejidad se debe a varias razones, entre ellas la representación de los datos. En el dataset del Titanic, los atributos están claramente definidos y son interpretables. Por ejemplo, las columnas representan atributos como sexo, edad, clase de viaje, etc. Estos atributos son intuitivos y fáciles de entender. En un dataset de imágenes, cada dato es un píxel con una intensidad específica, lo que significa que los atributos no son directamente interpretables en términos humanos. Otra de las razones podría ser la dimensionalidad, ya que los datos en tablas como la del Titanic suelen tener una dimensionalidad relativamente baja. En cambio, en imágenes, el preprocesamiento es más complejo e incluye tareas como la normalización de los valores de los píxeles, la reducción de ruido, el aumento de datos, etc. En conclusión, trabajar con imágenes añade una capa significativa de complejidad en cuanto a la interpretación, preprocesamiento y modelado de los datos, lo que hace que la exploración inicial sea más difícil en comparación con datasets tabulares como el del Titanic.

3. Clasificación binaria

Dada una imagen se desea responder la siguiente pregunta: **¿la imagen corresponde a la letra L o a la letra A?**

Para determinar si una imagen corresponde a la letra L o la A, primero seleccionamos del conjunto de datos original solo las imágenes de esas dos letras. Luego, verificamos la cantidad de imágenes por cada letra para ver si están balanceadas.

```
In [257]: asYls["Etiqueta"].value_counts()
Out[257]:
Etiqueta
A    2400
L    2400
```

Figura 5: A(50 %) y L (50 %).Están perfectamente balanceadas

Luego, dividimos el conjunto de datos en conjuntos de entrenamiento (80 %) y prueba (20 %) para poder entrenar y evaluar un modelo de clasificación de **KNN**. De esta manera, nuestro objetivo será poder asignar correctamente la letra A y L a cada imagen. Posteriormente, creamos una función en la que especificamos la cantidad de vecinos de **KNN** que deseamos utilizar calculando la exactitud **(accuracy)** para cada valor de vecinos, y una función a la cual le decimos la cantidad de atributos que queremos utilizar y nos devuelve los dataframe de train y test solo con la cantidad de atributos que le dijimos elegidos al azar. Esto nos permite crear gráficos para comparar diferentes configuraciones de **KNN** y diferentes cantidades de atributos. De esta manera, pudimos determinar cuál configuración es más conveniente utilizar.

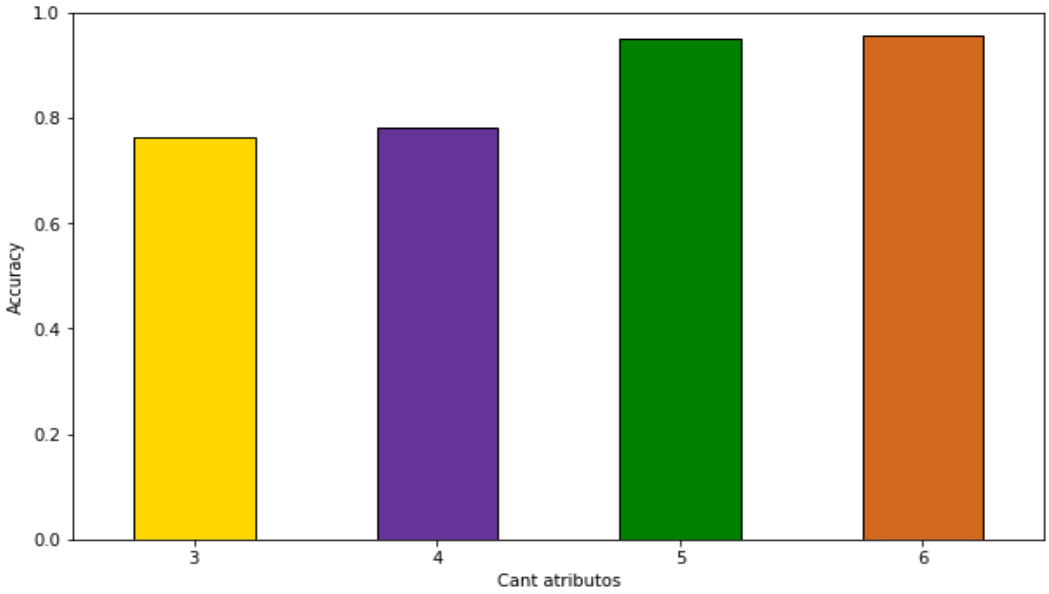


Figura 6: Accuracy entre modelos de knn con 6 vecinos y distinta cantidad de atributos

En la Figura 6, el gráfico de barras analiza la exactitud (accuracy) de modelos **KNN** utilizando k=6 vecinos más cercanos. Se comparan los resultados al usar 3, 4, 5 y 6 atributos seleccionados aleatoriamente. Se observa que al usar 5 o 6 atributos, el modelo **KNN** alcanza su mayor precisión. Esto sugiere que, en este caso particular, 5 o 6 atributos proporcionan la mejor representación de los datos para el modelo **KNN**. La precisión no mejora significativamente al pasar de 5 a 6 atributos, lo que podría indicar que el sexto atributo no añade información sustancial o incluso podría introducir ruido. Este análisis es crucial para entender cómo la selección del número de atributos afecta el rendimiento de modelos **KNN**. Seleccionar un número óptimo de atributos puede mejorar significativamente la precisión del modelo sin añadir complejidad innecesaria.

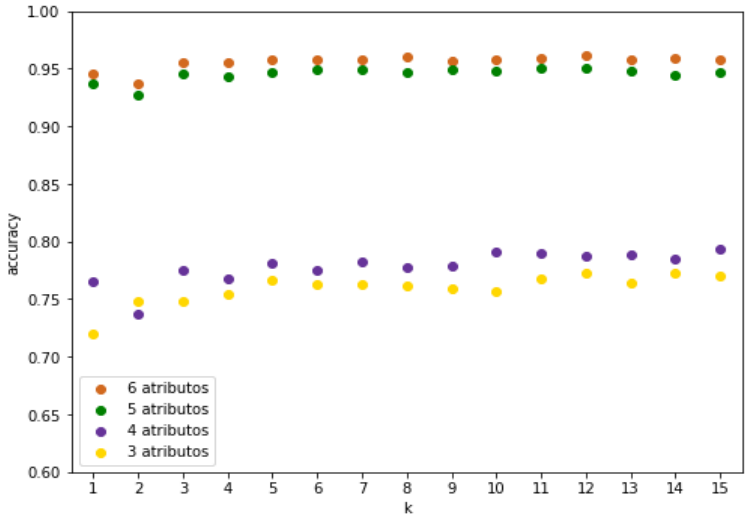


Figura 7: Accuracy entre modelos de knn con distinta cantidad de vecinos **(k)** y atributos

En la Figura 7, realizamos una comparación entre diferentes cantidades de atributos y vecinos más cercanos (KNN) con un scatter plot con el objetivo de determinar la mejor combinación en términos de exactitud. Se observa una diferencia notable al utilizar 3 o 4 atributos en comparación con 5 o más. Los resultados muestran que los mejores desempeños se logran al utilizar 6 atributos y 12 vecinos más cercanos, con una precisión aproximada del **0,961**. Es evidente que a partir de 5 atributos, las diferencias en el rendimiento son mínimas. Además, se observa que el número óptimo de vecinos no siempre es 12; por ejemplo, al utilizar 3 atributos, el mejor rendimiento se alcanza con 14 vecinos más cercanos en lugar de 12, y al utilizar 4 atributos, el mejor rendimiento se alcanza con 15 vecinos más cercanos en lugar de 12. Esto sugiere que la elección de la cantidad de vecinos depende de la configuración específica de atributos utilizados. Una observación importante es que los atributos se eligen de manera aleatoria, lo que significa que cuando hay pocos atributos, además de generar un modelo subajustado, la precisión depende en gran medida de la relevancia de los atributos seleccionados. Si el modelo elige aleatoriamente atributos no relevantes, es probable que su rendimiento no sea bueno.

4. Clasificación multiclase

Dada una imagen se desea responder la siguiente pregunta: **¿A cuál de las vocales corresponde la imagen?**

Para clasificar imágenes de las vocales A, E, I, O y U, seleccionamos las imágenes de esas letras del conjunto original y las dividimos en vocx y vocy y luego en vocx_dev, vocx_heldout, vocy_dev, vocy_heldout para luego entrenar un modelo de árbol de decisión con diferentes profundidades y seleccionar el mejor árbol usando validación cruzada con k-folding en el conjunto de vocx_dev y vocy_dev .

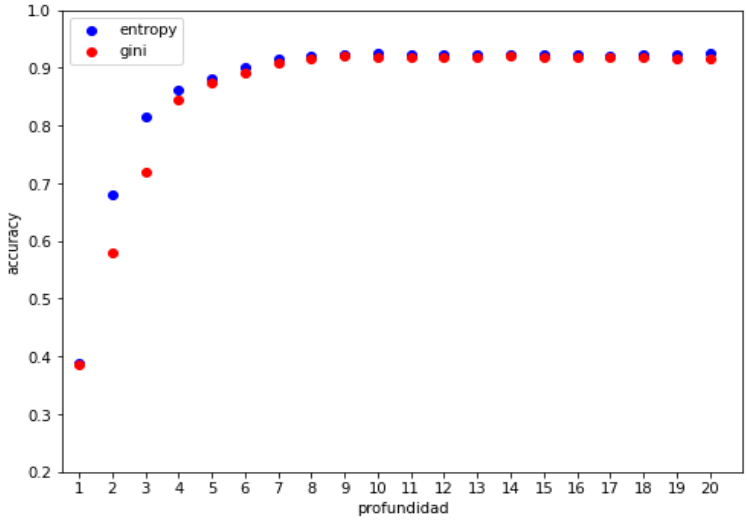


Figura 8: Accuracy entre diferentes hiperparàmetros

El objetivo del experimento fue ajustar y comparar diferentes modelos de árbol de decisión utilizando dos criterios distintos (entropía y Gini) y varias profundidades del árbol. Para evaluar el rendimiento de estos modelos, se utilizó la técnica de validación cruzada con k-folding, calculando el promedio de exactitud para cada configuración. Se creó una función para entrenar múltiples árboles de decisión variando el criterio de división (entropía y Gini) y la profundidad del árbol (de 1 a 20). Para ambos criterios, la exactitud aumenta con la profundidad del árbol hasta alrededor de una profundidad de 8. A partir de esta profundidad, no se observan mejoras sustanciales, indicando que profundidades mayores no aportan beneficios adicionales y podrían resultar en modelos más complejos sin ganancias en precisión. A profundidades menores (1 a 4), los árboles con criterio de entropía presentan una precisión ligeramente superior comparada con los árboles de criterio Gini. A partir de la profundidad 5, ambos criterios muestran un rendimiento muy similar, con precisiones que se acercan al 90 %. En general, no hay una diferencia significativa que favorezca a uno sobre el otro en profundidades mayores.

```
In [34]: runcell(18, '/h
('entropy', 10)
```

Figura 9: Mejores hiperparámetros

El mejor modelo en nuestro caso fue usando el criterio de entropía con la profundidad 10. Luego entrenamos este modelo elegido con los datos de todo el conjunto de desarrollo y lo usamos para predecir las clases en el conjunto held-out. Al hacer esto obtuvimos un score de 0.92625. Para analizar mejor la performance hicimos una matriz de confusión multiclase.

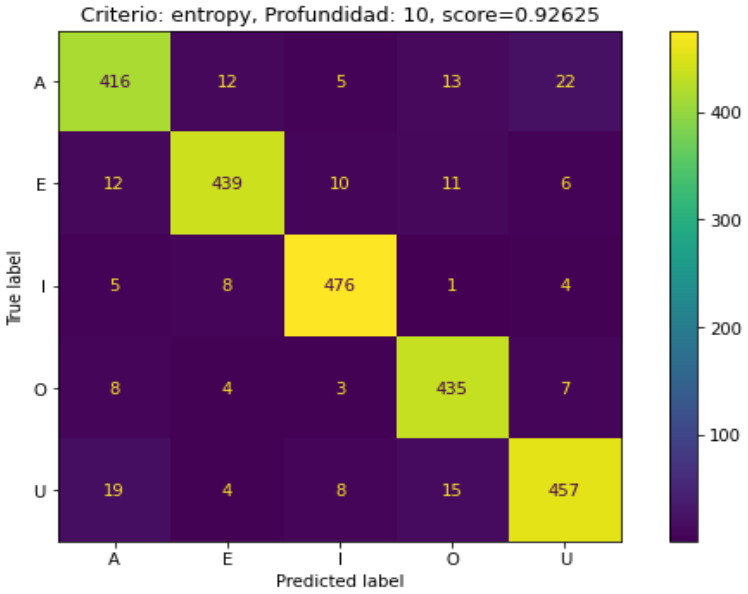


Figura 10: Matriz confusión

La matriz de confusión varía desde el color violeta hasta el amarillo, donde el violeta representa los valores más bajos y el amarillo los más altos. Con la métrica de entropía, encontramos la mejor performance con una profundidad de 10, lo que resulta en una matriz de confusión con un **92,6 %** de exactitud. La diagonal principal de la matriz tiene valores muy altos, lo que indica que la mayoría de las predicciones fueron correctas a pesar de tener algun que otro error visualizado en el resto de las posiciones. Se puede confirmar que estos errores no son relevantes ya que todos son de color violeta, el mas bajo posible, lo que indica que son poco frecuentes. Adicionalmente, se destaca que el modelo tiene una tasa de error muy baja al predecir la letra *I* con solo 18 errores y 476 aciertos. Por otro lado la letra mas predicha erroneamente es la *A* con 52 errores. Aunque hay algunos errores en la predicción, estos son mínimos y no afectan significativamente la alta precisión general del modelo.

5. Conclusión

En este trabajo práctico hemos abordado diversas técnicas de análisis y modelado para clasificar letras manuscritas utilizando el conjunto de datos EMNIST. A través de varios ejercicios aplicamos métodos de análisis exploratorio de datos, clasificación binaria y multiclase, y validación de modelos. El análisis exploratorio nos permitió entender la estructura y características del conjunto de datos, identificamos que los píxeles con tonalidades consistentes (amarillos y celestes) en las imágenes promedio de una letra son cruciales para la clasificación, mientras que los píxeles que permanecen en tonos violetas pueden ser considerados irrelevantes. Determinamos que ciertas letras comparten valores de píxeles similares, haciendo más difícil su diferenciación en comparación con otras. Utilizando el modelo KNN para hacer clasificación binaria con diferentes cantidades de atributos y vecinos más cercanos (k) pudimos comparar entre estos y, usando métricas de exactitud, quedarnos con el mejor modelo. Además, para clasificar entre las vocales (A, E, I, O, U), empleamos modelos de árboles de decisión y utilizamos validación cruzada con k-folding para evaluar diferentes configuraciones de hiperparámetros. Llegamos a la conclusión de que el mejor modelo fue un árbol de decisión con criterio de entropía y una profundidad de 10, luego de comparar con varios otros.

Este trabajo práctico ha demostrado la importancia del análisis exploratorio y la selección cuidadosa de modelos y parámetros para el éxito en tareas de clasificación. Los resultados obtenidos muestran que la selección de atributos y el ajuste de hiperparámetros son cruciales para mejorar la precisión del modelo, las técnicas de validación cruzada ayudan a evitar el sobreajuste y la clasificación de imágenes plantea desafíos, como la necesidad de preprocesamiento y la alta dimensionalidad de los datos, que requieren enfoques especializados en comparación con datos tabulares. En conclusión, hemos logrado construir y evaluar modelos eficaces para la clasificación de letras manuscritas, destacando la relevancia de una metodología rigurosa y un análisis detallado de los datos. Este ejercicio no solo reafirma los conceptos aprendidos en clase, sino que también proporciona una base sólida para abordar problemas de clasificación en escenarios más complejos en el futuro.