

IES Pere Maria Orts

Sistemas de Aprendizaje Automático

Práctica 2_2: Clasificación aprendizaje automático dataset CIFAR10

Autor:

Kenny Berrones

Profesor:

David Campoy Miñarro



iesperemariaorts



GENERALITAT
VALENCIANA

Índice

1. Introducción	2
2. Experimentos	3
3. Conclusiones	11

1. Introducción

CIFAR-10 es un conjunto de datos con imágenes etiquetadas en 10 clases (como aviones, perros, gatos, etc.). Este dataset lo usaremos empleando la librería Keras, con esta importaremos el dataset y podremos trabajar, en la siguiente imagen se puede observar ejemplos de imágenes de cada categoría:



Figura 1: Ejemplos de imágenes de cada categoría del dataset CIFAR-10

En la siguiente tabla vemos cada categoría con su respectivo índice:

Índice	Categoría
0	Avión (Airplane)
1	Automóvil (Automobile)
2	Pájaro (Bird)
3	Gato (Cat)
4	Venado (Deer)
5	Perro (Dog)
6	Rana (Frog)
7	Caballo (Horse)
8	Barco (Ship)
9	Camión (Truck)

Cuadro 1: Categorías del dataset CIFAR-10 y sus índices.

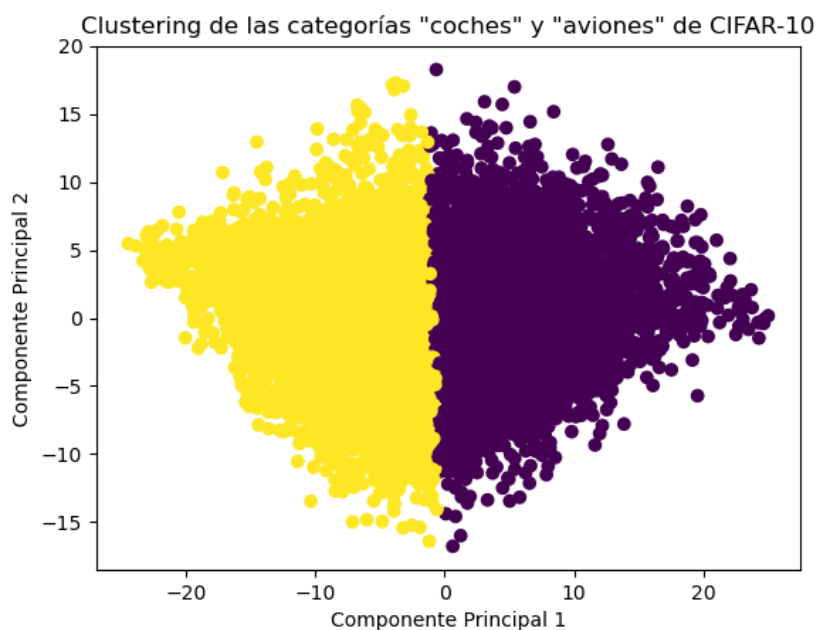
En esta práctica tenemos que utilizar una técnica llamada Clustering para entender como funciona. El clustering se basa en agrupar datos mediante características similares como puede ser colores, texturas entre otros. Se trata de una técnica que no pertenece al aprendizaje supervisado.

2. Experimentos

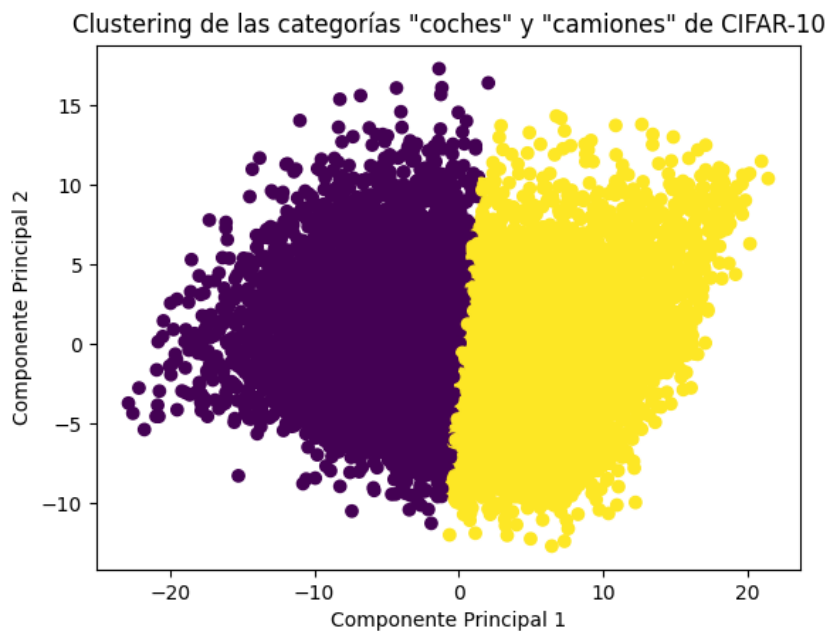
Vamos a realizar distintas pruebas con este dataset.

2.1. Clustering

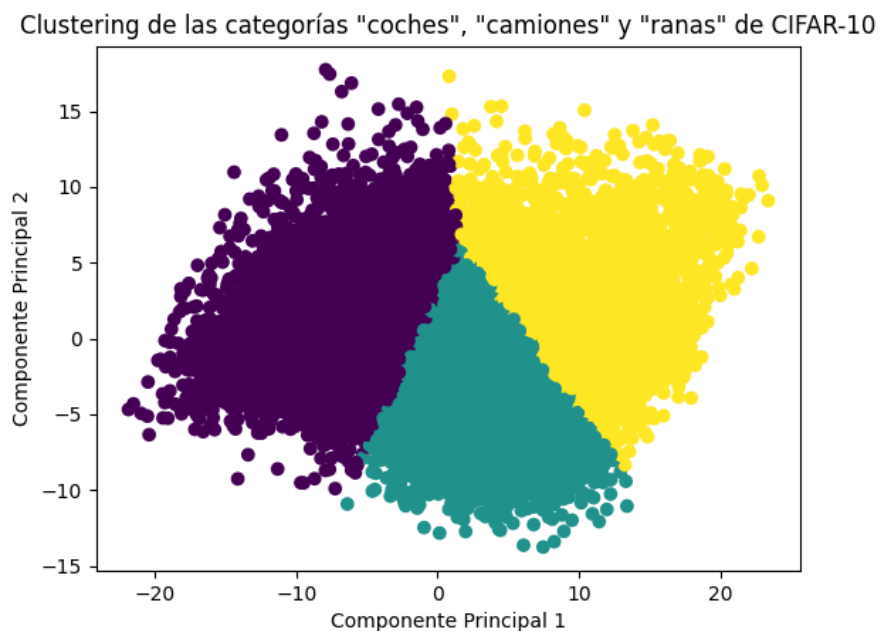
En primer lugar vamos a probar con el clustering, y vamos a probar para coches y aviones en primer lugar, obtenemos los siguientes resultados:



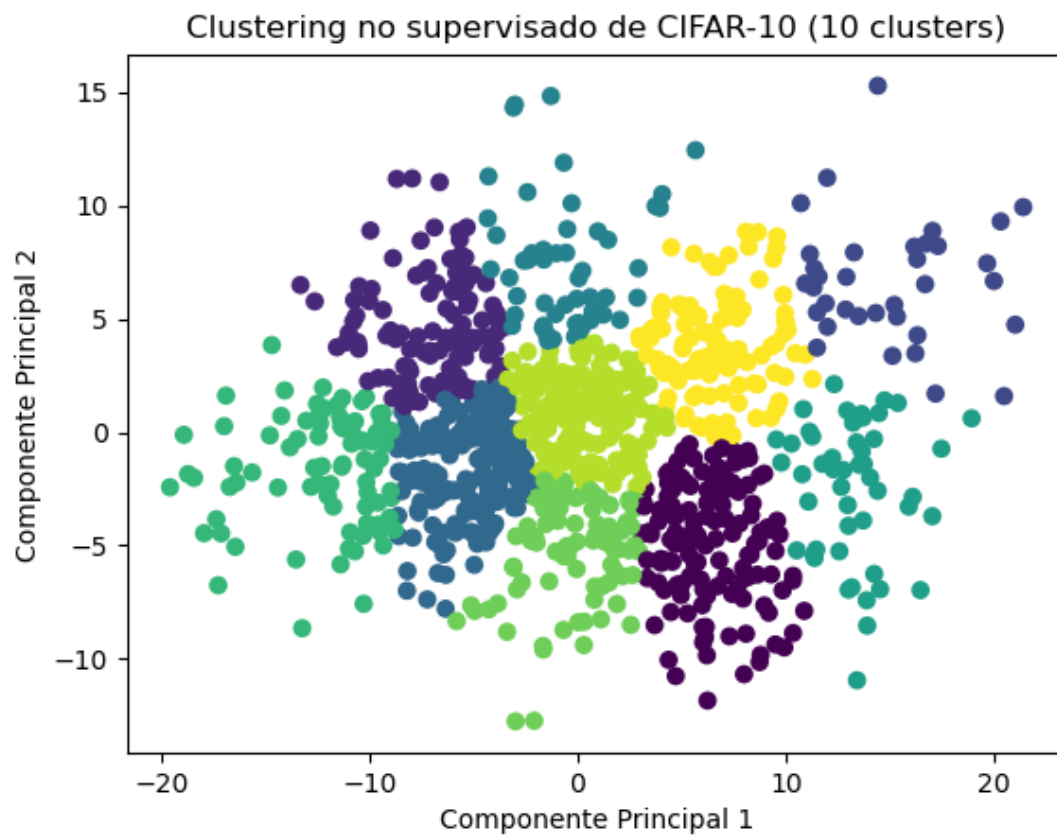
Ahora vamos a probar con coches y camiones, para ello tendremos que tocar el código que nos proporcionan:



Ahora vamos a probar con 3 categorías: camiones, coches y ranas, al igual que antes hemos tenido que modificar el código, los resultados son los siguientes:



Finalmente, en la segunda versión del código obtenemos estos resultados:



En la imagen anterior observamos 10 “categorías”, estas se corresponden con las 10 clases del dataset CIFAR-10.

2.2. Aprendizaje Automático supervisado

Ahora vamos a experimentar con el aprendizaje automático supervisado, tras ejecutar el código obtenemos el siguiente resultado:

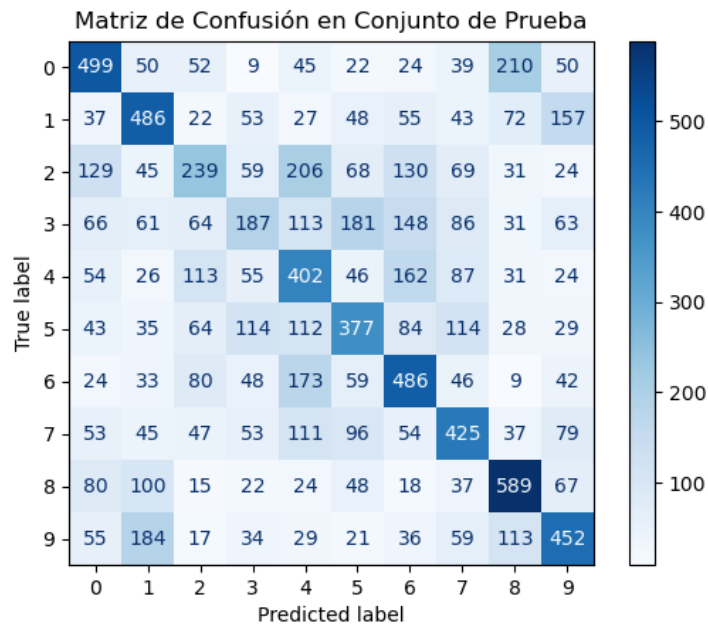


Figura 2: Matriz de confusión para Random Forest

De la matriz de confusión podemos sacar distintas conclusiones, pero la que más nos interesa es como se ha comportado este algoritmo, lo primordial es que nos tenemos que fijar en los colores de cada celda. En primer lugar, la diagonal se corresponden con los valores que son correctos, es decir, que la predicción y el valor real se corresponden, en las otras celdas lo contrario, pero además, si el color es más oscuro es que ha fallado más veces. Por ejemplo, si nos fijamos en la fila 0 y la columna 8 vemos que ha habido 210 ocurrencias donde la etiqueta real ha sido la clase 0 pero lo ha clasificado como la clase 8, esto se debe a que la forma de un barco y un avión pueden llegar a ser parecidas.

También vamos a apoyarnos en las siguientes estadísticas para ver como se ha comportado el modelo:

Informe de clasificación en el conjunto de prueba:				
	precision	recall	f1-score	support
0	0.48	0.50	0.49	1000
1	0.46	0.49	0.47	1000
2	0.34	0.24	0.28	1000
3	0.29	0.19	0.23	1000
4	0.32	0.40	0.36	1000
5	0.39	0.38	0.38	1000
6	0.41	0.49	0.44	1000
7	0.42	0.42	0.42	1000
8	0.51	0.59	0.55	1000
9	0.46	0.45	0.45	1000
accuracy			0.41	10000
macro avg	0.41	0.41	0.41	10000
weighted avg	0.41	0.41	0.41	10000

Figura 3: Estadísticas del algoritmo Random Forest

Si nos fijamos en la imagen anterior vemos que tanto la accuracy como el f1-score tienen un valor de 0.41, sin lugar a duda se trata de un valor muy pobre, por lo que el modelo no está generalizando correctamente y va a fallar en muchos casos.

2.3. Aprendizaje Automático NN

Ahora vamos realizar el aprendizaje automático con una red neuronal sencilla, obtenemos la siguiente tasa de acierto para este modelo:

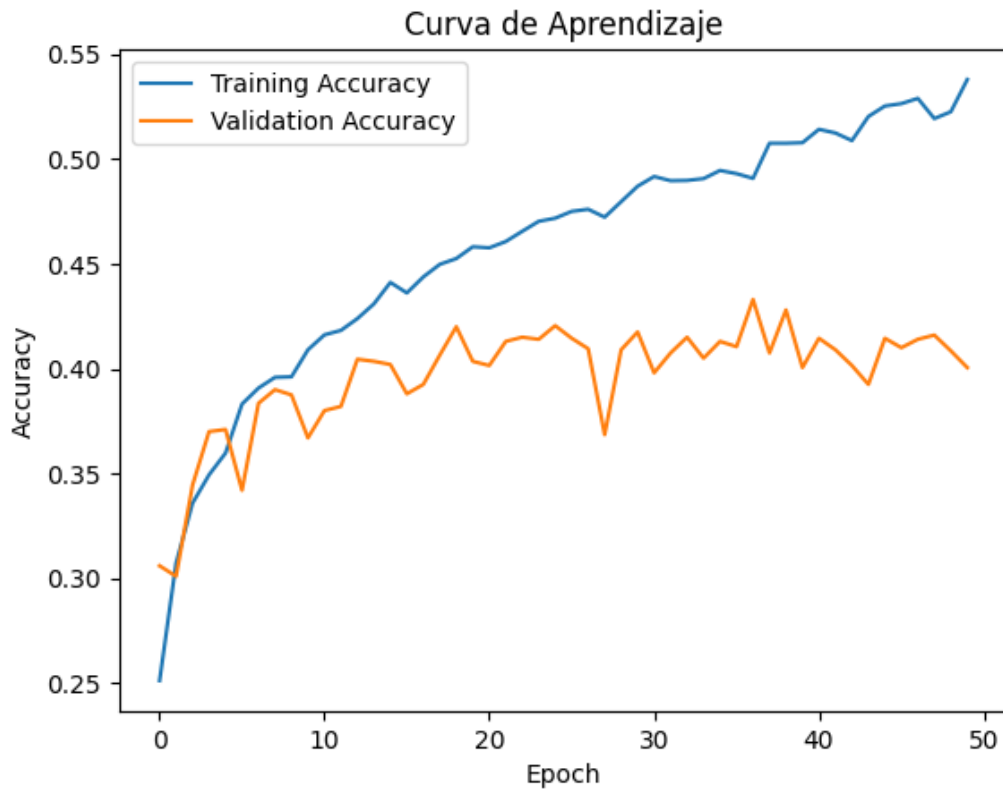


Figura 4: Tasa de acierto para los datasets de entrenamiento y validación

Vemos que la tasa de acierto es pobre también para este caso, pero vemos que con 50 épocas obtenemos una tasa de acierto para el dataset de entrenamiento de 0.55 y de 0.40 para el dataset de validación, quiero entender que estos valores irán incrementando a medida que aumentamos las épocas. Ahora obtenemos la matriz de confusión para este modelo:

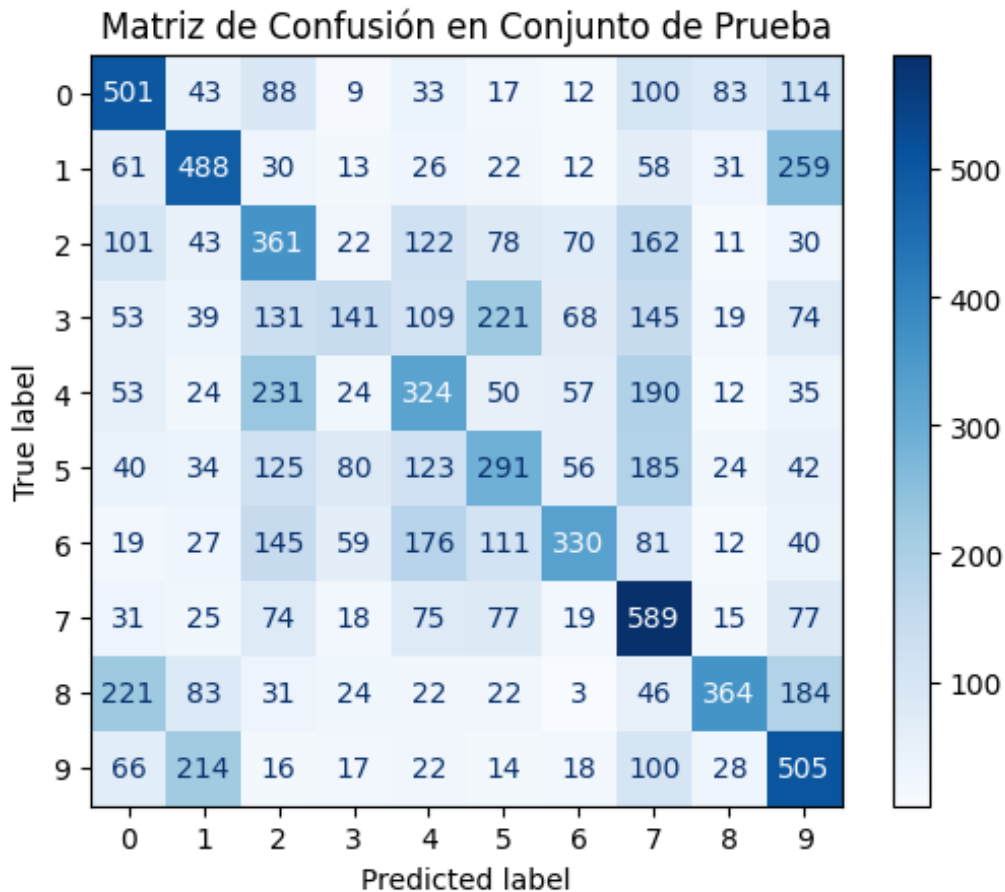


Figura 5: Matriz de confusión

Si observamos la gráfica anterior, vemos falla en muchas ocasiones, comparado con la matriz de confusión anterior podemos decir que más o menos son parecidas.

Para poder mejorar los resultados obtenidos podríamos aumentar las épocas de entrenamiento aunque esto podría llevar a un sobre entrenamiento, otra cosa que podríamos hacer es cambiar el modelo de la red neuronal, añadiendo capas ocultas, entre otras.

2.4. Aprendizaje Automático CNN

Finalmente, vamos a usar una **red neuronal convolucional (CNN)**, estas redes neuronales son muy eficientes para problemas de imágenes, ya que se aplican distintas convoluciones a las imágenes para que el modelo pueda aprender a lo largo de las distintas capas. En la siguiente imagen podemos apreciar un esquema de estas redes neuronales:

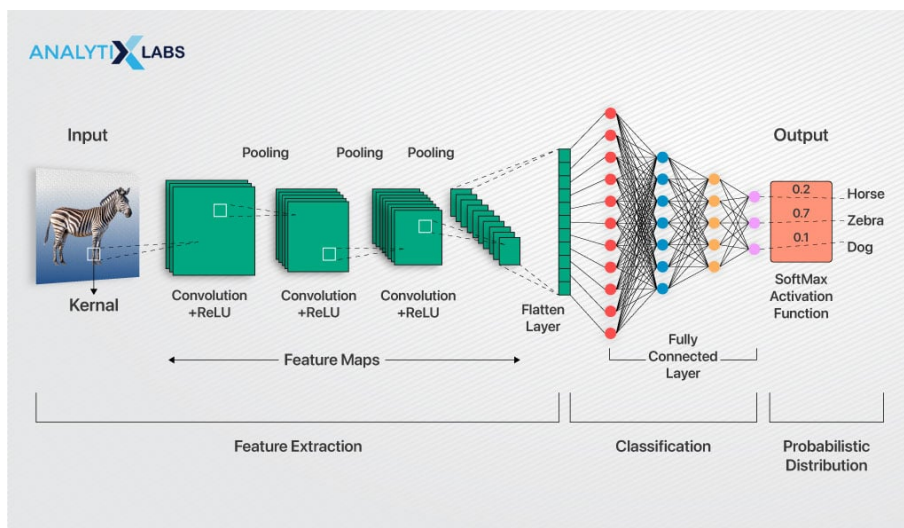


Figura 6: Esquema Redes Neuronales Convolucionales

Tras ejecutar el código proporcionado obtenemos los siguientes resultados:

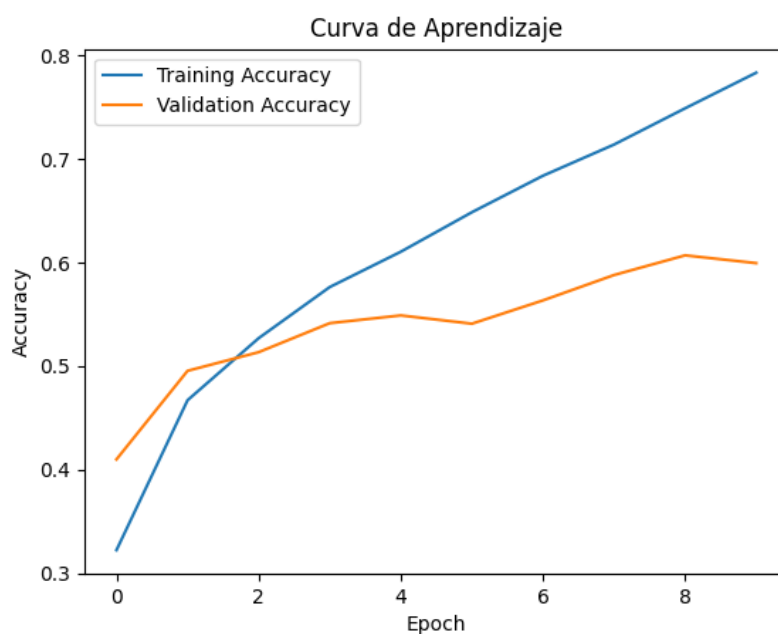


Figura 7: Tasa de acierto para los datasets de entrenamiento y validación

Vemos que obtenemos una tasa de acierto del 80 % para el dataset de entrenamiento con tan solo 10 épocas, para el dataset de validación obtenemos un acierto del 60 %. Ahora vamos a ver la matriz de confusión para el modelo, es la siguiente:

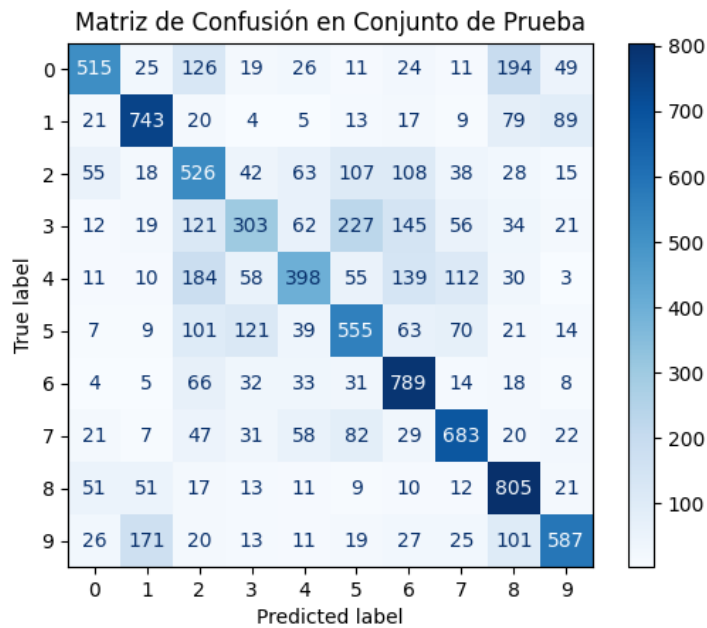


Figura 8: Matriz de confusión para el modelo CNN

En la gráfica anterior se puede apreciar que para ciertas clases funciona muy bien, pero para otras clases aún se sigue equivocando, por lo que con lo que hemos comentado se reduciría considerablemente este problema.

3. Conclusiones

En esta práctica se realizaron diversos experimentos con el dataset CIFAR-10 utilizando técnicas de clasificación tanto supervisadas como no supervisadas. En primer lugar, se exploró el uso de clustering para agrupar imágenes con características similares, lo cual permitió observar las limitaciones de esta técnica en categorías visualmente similares. Luego, se emplearon modelos de aprendizaje automático supervisado como Random Forest y redes neuronales básicas, donde la precisión de predicción fue baja, reflejando una falta de generalización. Finalmente, se implementó una red neuronal convolucional (CNN), la cual alcanzó un 85 % de precisión en el conjunto de validación, demostrando ser la técnica más efectiva en este contexto.