

Memoria del proceso de modelado Deep Learning

El objetivo de este trabajo es construir un modelo de Deep Learning capaz de predecir el nivel de engagement o la categoría principal de puntos de interés (POIs), combinando información visual (imágenes) y metadatos contextuales.

Metodología y evolución del modelo

Planteamiento inicial

He comenzado analizando el dataset, limpiando y preprocesando las variables (tratamiento de nulos, codificación de variables categóricas y numéricas, verificación de las rutas de imágenes, etc).

Aproximaciones previas

He realizado diferentes notebooks, cada uno suponiendo una iteración sobre el anterior:

- *Copia_de_PracticaDL_Beatriz_Velayos-1.ipynb:*

Primer intento de modelo híbrido, combinando una red neuronal simple para metadatos y un extractor de características de imágenes (ResNet18). Se encontraron problemas con la preparación del input, la normalización y la integración de ambas ramas. Los resultados iniciales fueron bajos, con dificultades para el entrenamiento conjunto.

- *Copia_de_PracticaDL_Beatriz_Velayos-2.ipynb:*

Se mejoró la integración de imágenes y metadatos, ajustando la arquitectura y la función de pérdida. Sin embargo, el modelo sufría de overfitting rápido y las métricas de validación eran inestables. Se detectaron problemas de desbalanceo de clases y dificultad para generalizar.

- *PracticaDL_Beatriz_Velayos_3.ipynb:*

Se abordaron los problemas anteriores aplicando regularización, técnicas de data augmentation sobre imágenes y balanceo de clases. Se mejoró la preparación del dataloader y se ajustaron los hiperparámetros. El rendimiento mejoró ligeramente pero seguía habiendo margen de mejora, especialmente para clases minoritarias.

- *PracticaDL_Beatriz_Velayos-4.ipynb:*

En esta versión se pulieron detalles del preprocesamiento y de la integración del pipeline. El código se hizo más modular y se refinaron métricas e interpretación de resultados. El rendimiento fue más consistente, pero se observaba que para algunas clases el modelo seguía sin funcionar correctamente.

Descripción del modelo final

El modelo final, desarrollado en PracticaDL_Beatriz_Velayos.ipynb, consiste en un pipeline híbrido: una rama basada en ResNet18 preentrenada para procesar imágenes, y otra rama de fully connected layers para los metadatos. Ambas salidas se concatenan y pasan por una capa final para la predicción. Se emplea validación cruzada, métricas detalladas y análisis de errores. El preprocesamiento se ha optimizado y se han aplicado técnicas de regularización y data augmentation para mejorar la generalización. Aunque el accuracy global no es alto, la solución es coherente y sólida respecto a las limitaciones del dataset.

Resultados obtenidos

El modelo final alcanza una accuracy global del 34% y un F1 macro de 0,17 sobre el conjunto de test. Esto indica que el modelo identifica correctamente una de cada tres muestras, aunque su capacidad para distinguir las clases minoritarias sigue siendo baja. El análisis de las métricas muestra que el rendimiento es aceptable para clases frecuentes, pero insuficiente en aquellas con pocos ejemplos. Durante el proceso se detectaron problemas de desbalanceo y dificultad para que el modelo capture la variabilidad entre categorías, a pesar de las mejoras en el pipeline.

Conclusiones

El trabajo ha permitido desarrollar un pipeline funcional de Deep Learning multimodal, capaz de integrar imágenes y metadatos para la clasificación de POIs.

El resultado final, aunque lejos de la perfección, constituye una base reproducible sobre la que seguir investigando.

Posibles mejoras

- Probar arquitecturas de imagen más avanzadas (ResNet50, EfficientNet) o técnicas de fine-tuning más agresivas.
- Aplicar técnicas de oversampling/undersampling o focal loss para mitigar el desbalanceo de clases.
- Explorar modelos de atención (attention pooling) para fusionar mejor las ramas de imagen y metadatos.
- Investigar la explicación de predicciones (ej. Grad-CAM) y análisis de errores para depurar la clasificación de clases minoritarias.
- Recopilar más datos o aplicar técnicas de generación de datos sintéticos para aumentar la representación de clases poco frecuentes.
- Ajustar hiperparámetros y experimentar con otros optimizadores o técnicas de aprendizaje (One Cycle, learning rate scheduling, etc.).