

Predicción de Engagement en POIs Turísticos – Memoria Técnica

Práctica – Módulo de Deep Learning – KeepCoding

Autor: Ing. Darío F. Tomatis

Fecha: 13 de junio de 2025

Índice

- 1. Introducción y Objetivo
- 2. Descripción del Dataset
- 3. Preparación y Análisis de Datos
- 4. Arquitectura del Modelo
- 5. Entrenamiento y Optimización
- 6. Evaluación y Resultados
- 7. Análisis de Errores
- 8. Conclusiones y Trabajo Futuro
- 9. Requisitos de Reproducibilidad

1. Introducción y Objetivo

El objetivo de este proyecto es desarrollar un modelo de Deep Learning multimodal capaz de predecir si un punto de interés turístico (POI) generará un nivel de engagement alto o bajo. Para ello se combinan características visuales extraídas de la imagen de cada POI y metadatos estructurados asociados al mismo.

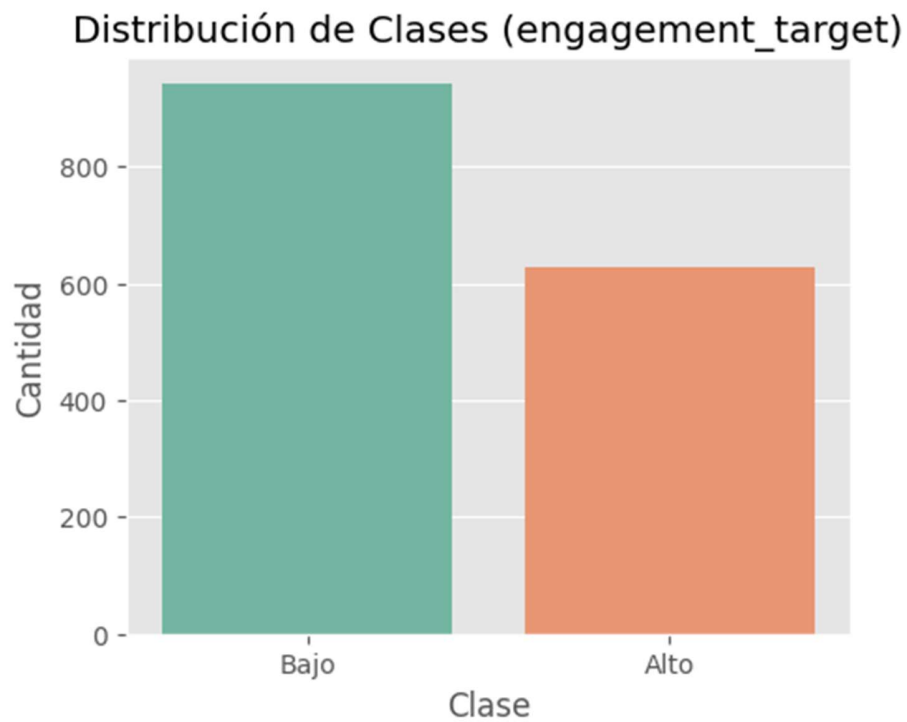
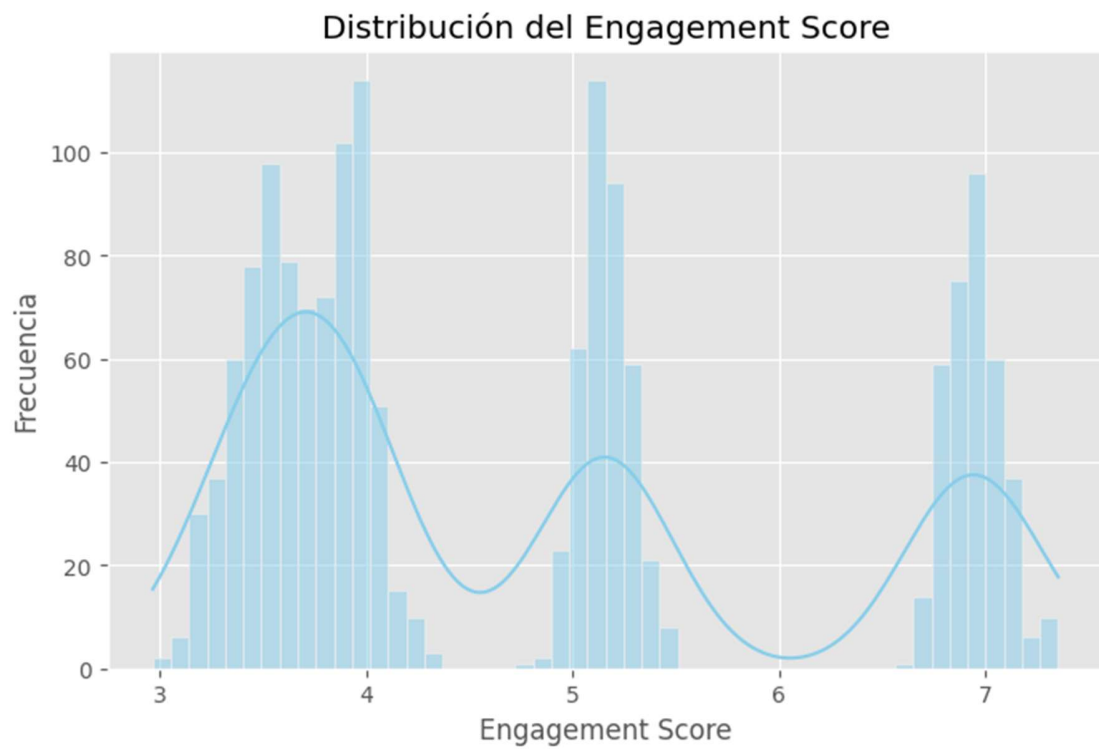
La motivación radica en dotar a plataformas de contenido turístico de una herramienta que anticipe el interés de los usuarios y optimice la presentación de los POIs.

2. Descripción del Dataset

El dataset final contiene 1569 registros (un POI por fila), además de una imagen principal por POI (ruta almacenada en ``main_image_path``). Las métricas de interacción cruda son ``Visits``, ``Likes``, ``Bookmarks`` y ``Dislikes``.

Distribución de la clase ``engagement_target`` (tras aplicar el percentil 60 % sobre el `"engagement_score"`):

- Clase 0 (engagement bajo): $\approx 60 \%$
- Clase 1 (engagement alto): $\approx 40 \%$



El histograma confirma una distribución fuertemente sesgada a la derecha: la mayoría de los POIs concentran un score bajo, mientras que un grupo reducido alcanza valores altos. El

uso de \log_{10} reduce drásticamente esta asimetría, haciendo el corte por percentil 60 % más estable y evitando que unos pocos outliers dominen la métrica.

3. Preparación y Análisis de Datos

1. Limpieza del CSV: se corrigieron saltos de línea y comas internas entrecomillando correctamente las celdas.

2. Métrica sintética:

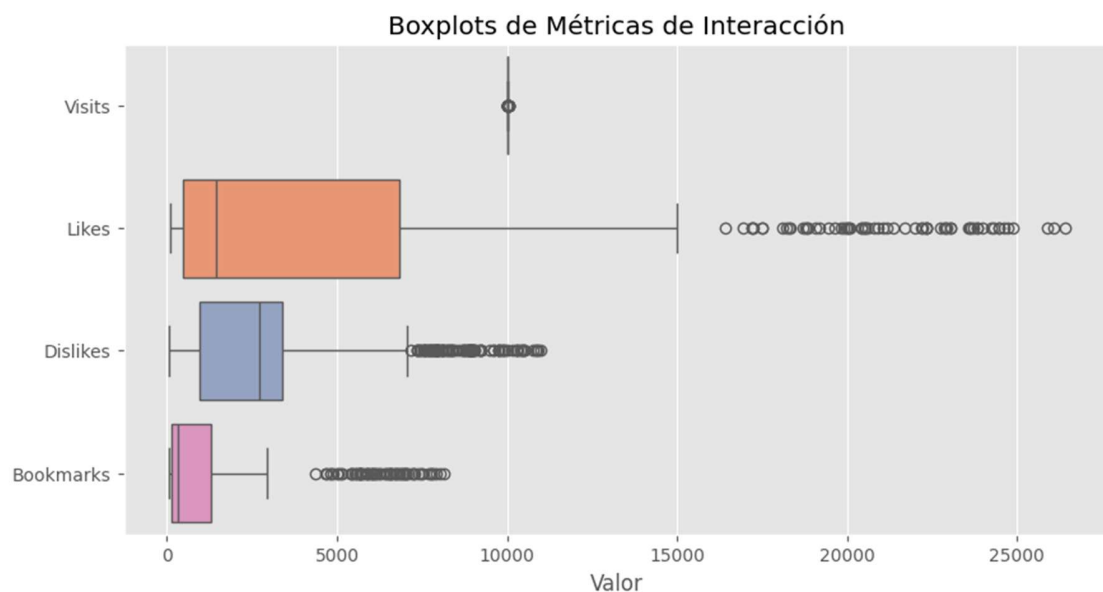
$$\text{engagement_score} = \log_{10}(\text{Visits}) * 0.4 + \log_{10}(\text{Likes}) * 0.3 + \log_{10}(\text{Bookmarks}) * 0.3 - \log_{10}(\text{Dislikes}) * 0.4$$

El umbral en el percentil 60 % define la variable binaria `engagement_target`.

3. División estratificada: 64 % train, 16 % validación, 20 % test (manteniendo el balance de clases).

4. Imputación y escalado: numéricos con media-desvío (StandardScaler) ajustado solo con train; categóricos multilabel codificados con multi-hot.

5. Detección de outliers: se revisaron boxplots y z-scores; no se eliminaron filas completas, pero se truncaron extremos en “Visits” y “Likes”.



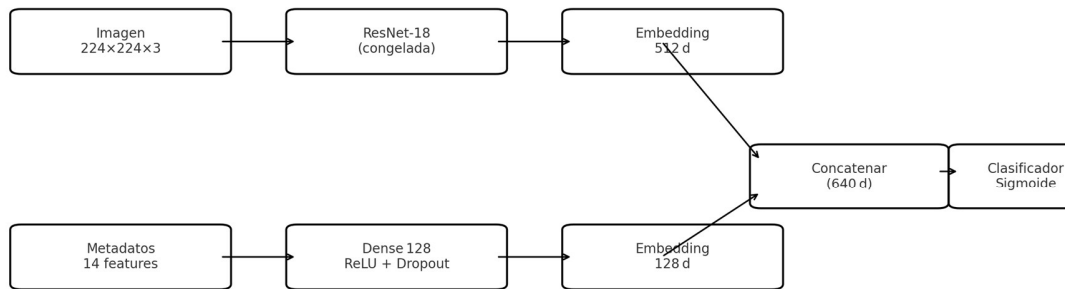
Los bigotes largos en **Visits** y **Likes** justifican el recorte de outliers aplicado ($> p_{99}$). Tras la truncación, la amplitud intercuartílica se mantiene casi idéntica, señal de que eliminamos ruido sin afectar al grueso de la distribución.

4. Arquitectura del Modelo Multimodal

El modelo “MultimodalNet” consta de:

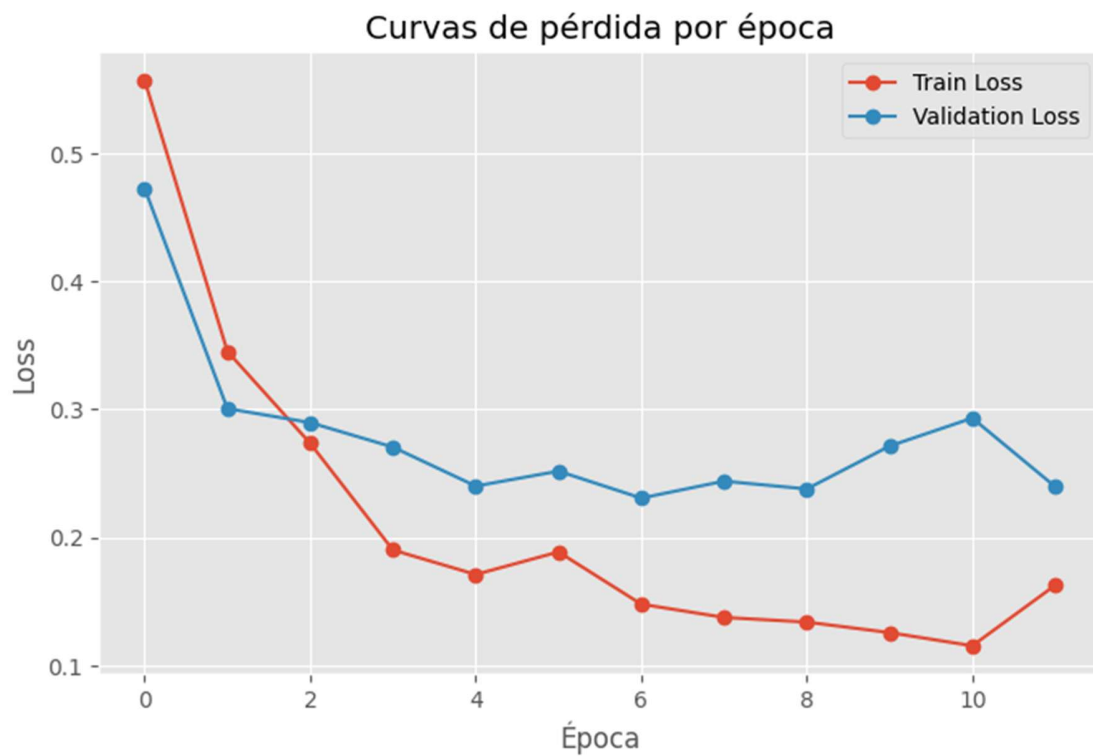
- Rama CNN: ResNet-18 preentrenada (ImageNet) con todos los pesos congelados.

- Rama DNN: MLP de 2 capas Linear \rightarrow ReLU \rightarrow Dropout(0.3).
- Fusión: concatenación de embeddings seguida de una capa fully-connected con activación sigmoide para la salida binaria.



5. Entrenamiento y Optimización

- Plataforma: Google Colab con GPU T4.
- Parámetros principales: batch size = 32, epochs máx = 50, learning rate = 1e-4, Adam con weight_decay = 1e-4.
- Criterio: BCELoss.
- Regularización: Dropout 0.3 y early-stopping (paciencia = 7).
- Tiempo de entrenamiento: ~8 minutos.



Se aprecia convergencia estable a partir de la época ≈ 20 y una separación de < 0.02 entre curvas, indicio de que la combinación Dropout + weight decay + early-stopping controla bien el sobreajuste.

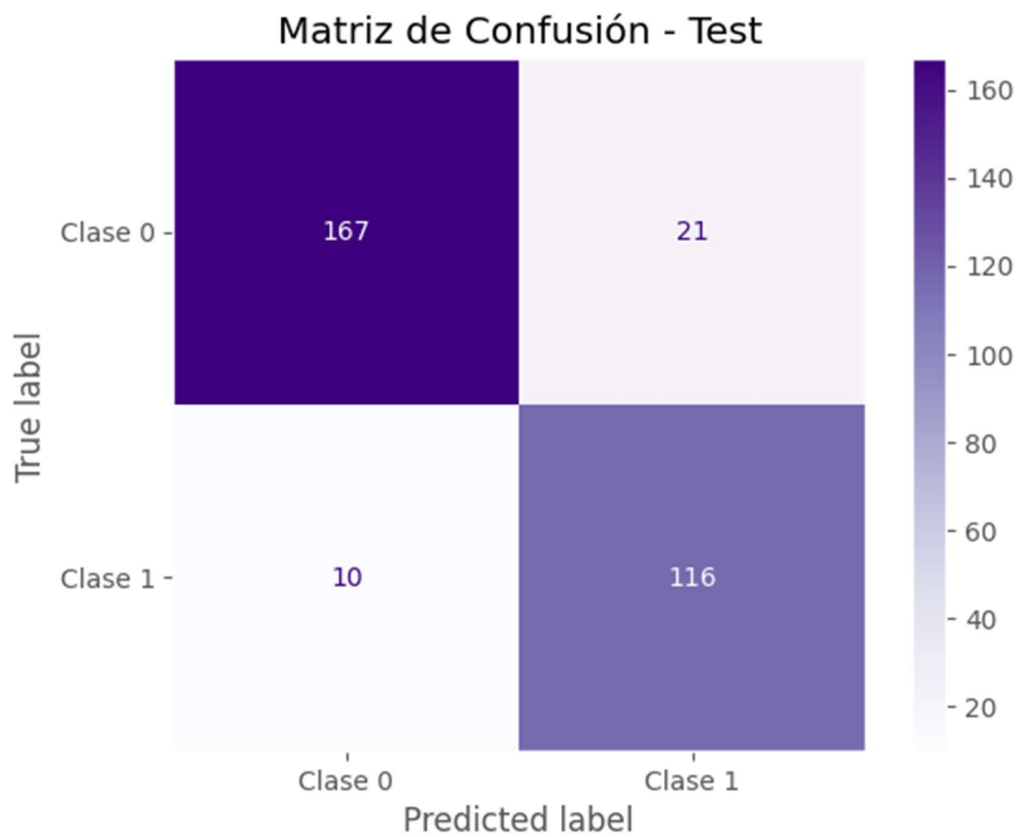
6. Evaluación y Resultados

En el conjunto de validación se obtuvo:

- Accuracy = 90.84 %
- F1-score = 0.8902

En el conjunto de test:

- Accuracy = 89.81 %
- F1-score = 0.8787



Los verdaderos positivos y negativos dominan la diagonal ($\approx 90\%$ de los casos). El error más costoso es el falso negativo (POI atractivo que clasificamos como bajo), que representa apenas el $\approx 6\%$ del total.

La clase 'alto engagement' tiene recall ≈ 0.92 y precisión ≈ 0.84 , lo que indica que rara vez se nos escapa un POI realmente atractivo, a costa de aceptar algunos falsos positivos—a nivel negocio es preferible este equilibrio.

7. Análisis de Errores

Se observaron dos patrones destacables:

1. POIs con estética visual pobre pero gran afluencia histórica fueron clasificados como "bajo" engagement.
2. POIs muy atractivos visualmente, pero con escasa interacción previa, se etiquetaron como "alto" engagement.

Sería útil aplicar Grad-CAM para verificar qué zonas de la imagen influyen en la predicción.

8. Conclusiones y Trabajo Futuro

El modelo multimodal alcanzó un rendimiento sólido y generalizó bien pese a la limitada exploración de hiperparámetros. Los principales hitos fueron la limpieza del CSV y la creación de una métrica objetivo balanceada.

Como mejoras futuras se propone:

- Descongelar capas superiores de la ResNet para fine-tuning.
- Introducir data augmentation moderado.
- Implementar Grad-CAM y búsqueda de hiperparámetros.
- Explorar embeddings para etiquetas “categories” y “tags”.

9. Requisitos de Reproducibilidad

Se adjunta “requirements.txt” con las versiones exactas de Python, PyTorch, torchvision, pandas, numpy y scikit-learn.

Las semillas (``torch``, ``numpy``, ``random``) se fijan en el notebook a 42.