

# Memoria Técnica - Práctica Final

## ✦ Proyecto: Predicción del Éxito de Atracciones Turísticas

---

### ”If Objetivo General

Desarrollar un modelo de deep learning capaz de predecir si un punto de interés turístico (POI) generará un alto o bajo nivel de engagement. Para ello, se han utilizado tanto sus atributos visuales (imagen principal) como metadatos estructurados (visitas, likes, descripciones, etc.).

---

### Dataset

El dataset contiene información de 1.569 POIs con las siguientes características: - Imagen principal (main.jpg) por cada POI. - Metadatos: identificador, nombre, descripción, categorías, número de visitas, likes, dislikes, bookmarks. - Métrica calculada de engagement: suma de interacciones positivas menos negativas. - Clase binaria engagement: alto (1) o bajo (0) en función de la mediana del score.

---

### Preparación y Preprocesamiento

1. **Carga desde Google Drive:** Los datos ya estaban descomprimidos.
  2. **Verificación de integridad:** Se validó que cada POI tenga su imagen.
  3. **Creación de variables clave:**
    - engagement\_score: Suma de Likes + Bookmarks + Visits - Dislikes.
    - engagement: Etiqueta binaria basada en la mediana.
    - has\_image: Si existe la imagen asociada al POI.
  4. **División estratificada:** En conjuntos de entrenamiento (70%), validación (15%) y test (15%).
  5. **Estandarización:** Solo de las variables numéricas del conjunto estructurado.
- 

### Modelo 1 - CNN + MLP Propio

#### Arquitectura

- CNN personalizada con capas Conv2D, ReLU, MaxPool.
- MLP para metadatos con capas lineales, Dropout y ReLU.
- Concatenación final y capa Sigmoid para clasificación binaria.

### Entrenamiento

- Función de pérdida: BCELoss y BCEWithLogitsLoss (ajustada posteriormente).
- Optimizador: Adam con  $lr=1e-4$  -  $0.01$ .
- Estrategias:
  - Reducción del batch size ( $32 \rightarrow 16$ ).
  - Aumento de Dropout ( $0.25 \rightarrow 0.5$ ).
  - Ajuste de epochs entre 6 y 15.

### Resultados

- Accuracy en test: **0.96**
  - Matriz de confusión:
    - Predicciones correctas tanto para engagement bajo como alto.
  - Conclusión: modelo simple pero eficaz. Menor sobreajuste.
- 

## Modelo 2 - Multimodal con ResNet18

### Arquitectura

- ResNet18 preentrenada (sin capa final) para las imágenes.
- MLP para datos estructurados.
- Capa combinada con Sigmoid al final.

### Entrenamiento

- Mismo esquema y parámetros iniciales que el modelo 1.

### Resultados

- Accuracy en test: **0.54**
  - Alta cantidad de falsos positivos.
  - Conclusión: el modelo no aprendió correctamente. El uso de una arquitectura compleja sin un entrenamiento adaptado provocó sobreajuste y mala generalización.
- 

## Reproducibilidad

- Se definió SEED=42 para random, numpy, torch y torch.cuda.
  - Se generó requirements.txt con las versiones exactas de las dependencias.
- 

## Conclusiones Finales

- Un modelo simple bien ajustado puede superar a arquitecturas complejas.
  - El entrenamiento cuidadoso y el preprocesamiento correcto fueron clave.
  - El modelo 1 demostró mejor generalización.
-

## Entregables Incluidos

- notebook\_practica\_final.ipynb
  - best\_model.pt(pertenece al modelo 2)
  - requirements.txt
  - memoria\_tecnica.pdf
- 

## Reflexión Final

Este proyecto demostró la importancia de entender los datos, ajustar adecuadamente el modelo y analizar los resultados de forma crítica. La flexibilidad, el aprendizaje por prueba y error, y la interpretabilidad fueron clave en la elección del mejor modelo.